Interpretable Deep Learning

AI602: Recent Advances in Deep Learning

Lecture 16

Slide made by

Jun Hyun Nam

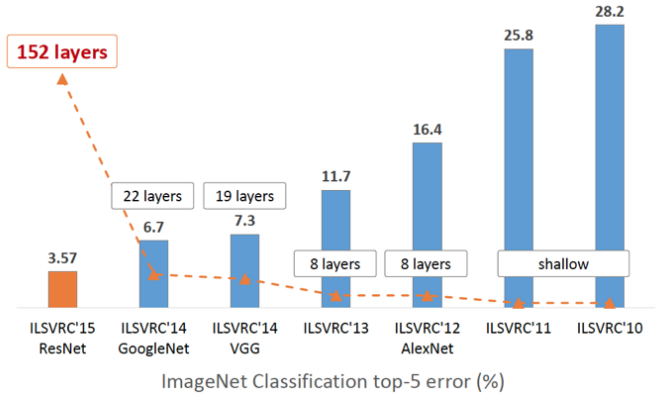
KAIST EE

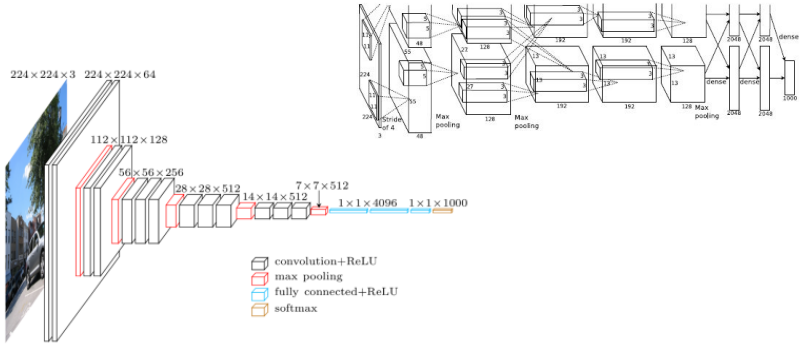
Table of Contents

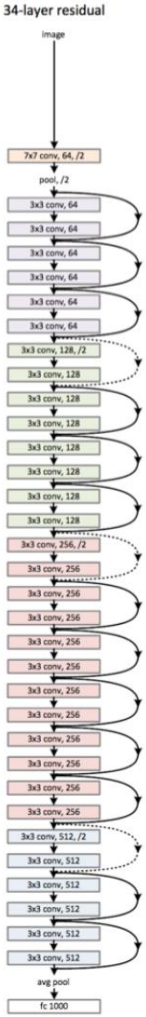
Table of Contents

Why do we need interpretability?

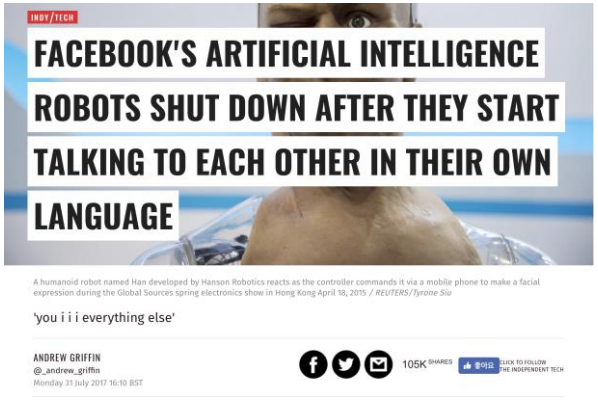
- **Recent deep learning models are too complex to understand**
  - Deep learning shows dramatically improved performance on various tasks (e.g. image classification, object detection, visual question answering)
  - Superior performance rely on deep and complex architecture

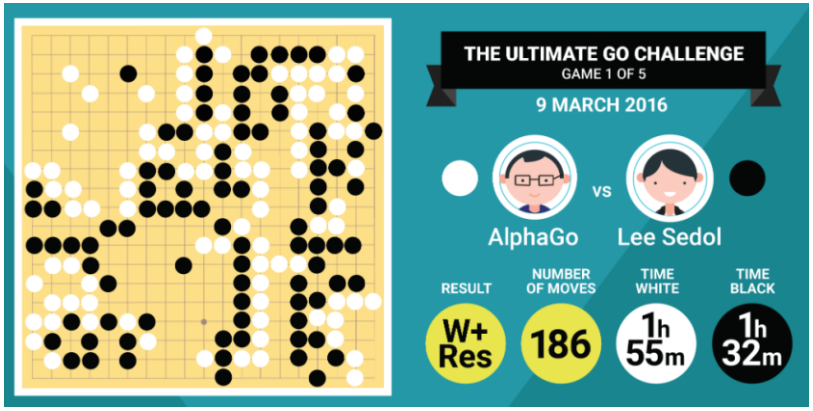






- We want to **understand** what's going on in the black-box model





When do we need interpretability?

- We **don't need** interpretability for *every single model*
  - No significant consequences for unacceptable results (e.g. recommendation system)
  - The problem is sufficiently well-studied and validated (e.g. postal code sorting)
- We need interpretability for **reliable model**
  - Safety critical domains requires reliability for decision making
  - User should be understand the *internal decision making process*

- We need interpretability for **scientific understanding**
  - Human want to understand *super-human performance* for various tasks
  - e.g. image recognition, AlphaGo
  - Not a main focus of this lecture

What is interpretability?

- **Definition of interpretability**

  "The *ability to explain* or to present in *understandable terms* to a **human**"

- **What is NOT interpretability?**
- Interpretability is not about making all models interpretable
  - There are many applications that don't need interpretability
  - e.g. advertisement, recommendation system
- Interpretability is not about understand every single bit of the model
  - We don't need to understand internal mechanism of computer to use it
  - We only need *high-level description* about how it works
- Interpretability is not against developing highly complex models
  - Most of the successful models are highly complex
  - We don't need to redevelop from the scratch

Overview of Lecture: Interpretability Methods

- **Feature attribution**
  - *Which part of the input affected the prediction?*

Which features of the mushroom make it model to predict that it is edible?

Which part of the image make it model to predict the image as dog(cat)?

- **Human-aligned concept**
  - *Does the neural network reflect human knowledge?*

human knowledge

= or ≠

neural network representation

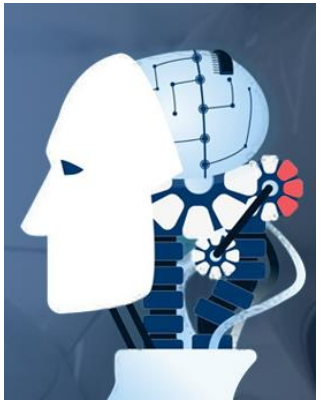
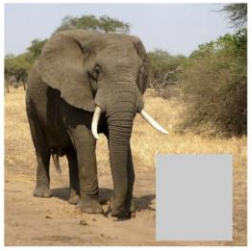
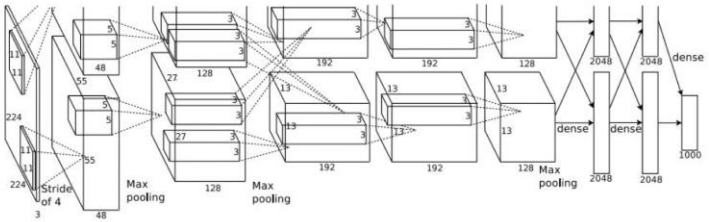


Table of Contents

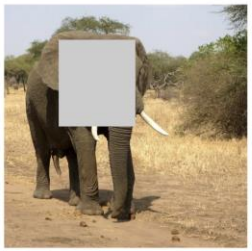
Occlusion Map [Zeiler et al., 2014]

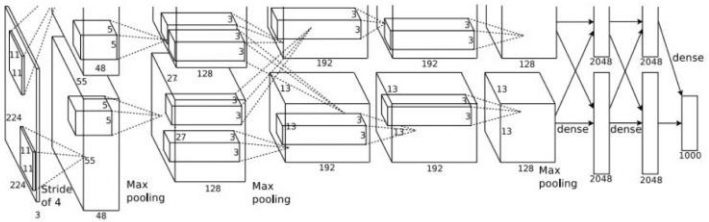
- **Idea**: Mask part of the image with gray patch before feeding to CNN, and check how much the prediction changes





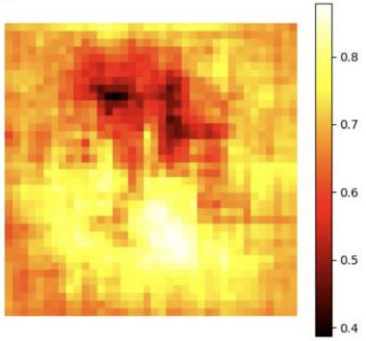
P(elephant) = 0.95



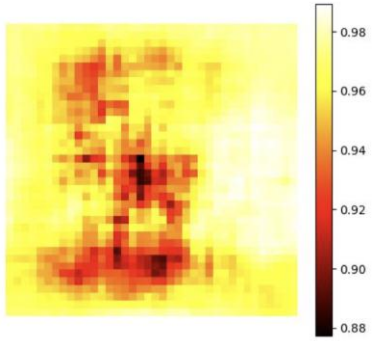


P(elephant) = 0.75

African elephant, Loxodonta africana



schooner



Prediction Difference Analysis [Zintgraf et al., 2017]

- **Problem**: Removing information with gray patch is too heuristic
- **Idea**: Simulate the absence of a feature by marginalizing the feature

- **Goal**: The attribution of i-th feature for given image and $\mathbf{x}$ and class c

$$p(c|\mathbf{x}) - p(c|\mathbf{x}_{\setminus i})$$

where $\mathbf{x}_{\setminus i}$ represents the absence of x_i in $\mathbf{x}$

$$p(c|\mathbf{x}_{\setminus i}) = \sum_{x_i} p(x_i|\mathbf{x}_{\setminus i}) p(c|\mathbf{x}_{\setminus i}, x_i)$$

- Note that $p(x_i|\mathbf{x}_{\setminus i})$ is computationally expensive

- Assume x_i is independent of the other features, i.e., $p(x_i|\mathbf{x}_{\setminus i}) \approx p(x_i)$

$$p(c|\mathbf{x}_{\setminus i}) \approx \sum_{x_i} p(x_i) p(c|\mathbf{x}_{\setminus i}, x_i)$$

- The prior probability $p(x_i)$ is usually approximated by the empirical distribution

Prediction Difference Analysis [Zintgraf et al., 2017]

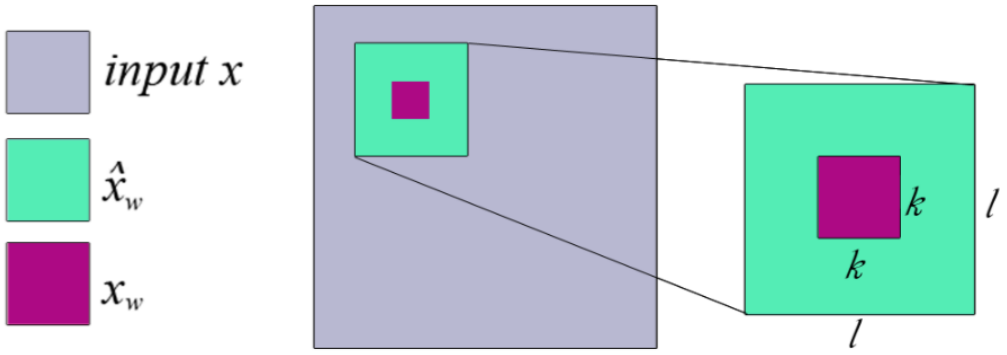
- **Idea**: Simulate the absence of a feature by marginalizing the feature

$$p(c|\mathbf{x}_{\setminus i}) = \sum_{x_i} p(x_i|\mathbf{x}_{\setminus i}) p(c|\mathbf{x}_{\setminus i}, x_i)$$

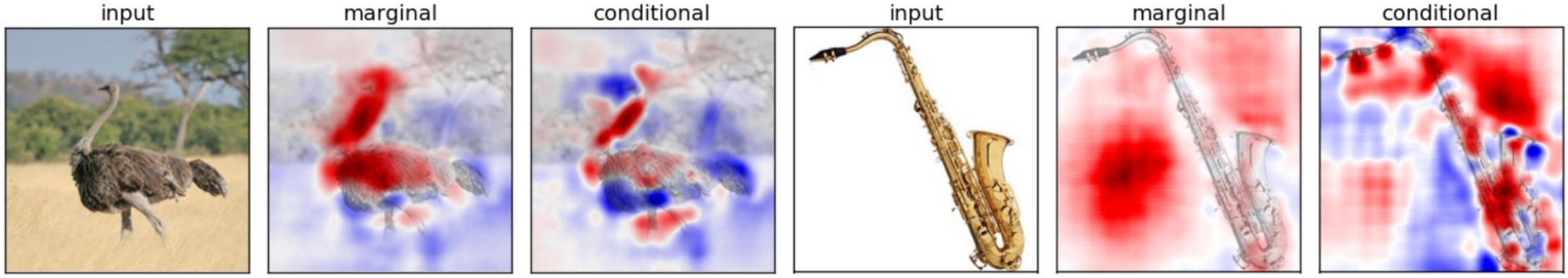
- **Problem**: $p(x_i|\mathbf{x}_{\setminus i}) \approx p(x_i)$ is a very crude approximation
  - e.g. a pixel's value is highly dependent on other pixels

- **Observations**
  - A pixel depends most strongly on a small neighborhood around it
  - The conditional of a pixel given its neighborhood does not depend on the position

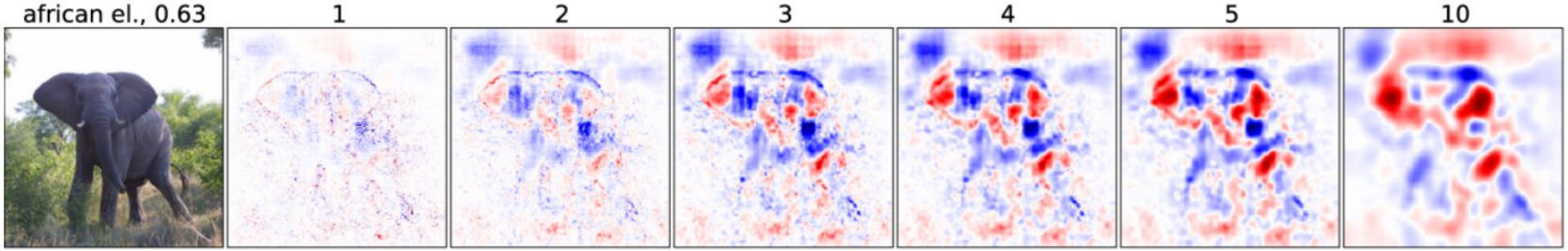
- For a pixel x_i, one can find a patch $\hat{\mathbf{x}}_i$ than contains x_i and $p(x_i|\mathbf{x}_{\setminus i}) \approx p(x_i|\hat{\mathbf{x}}_i)$



- **Results**
  - Marginal vs. conditional sampling

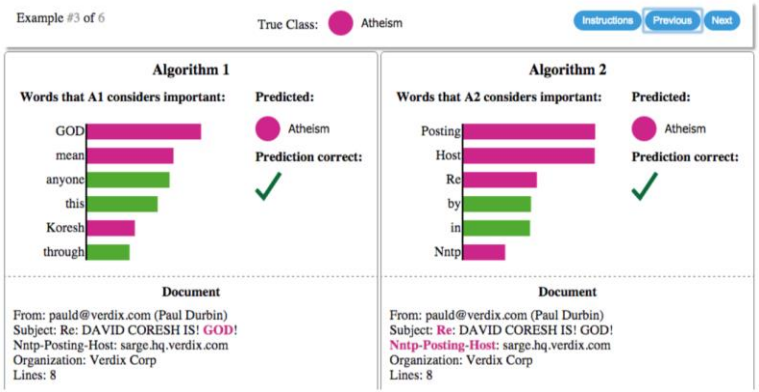


  - Different window sizes



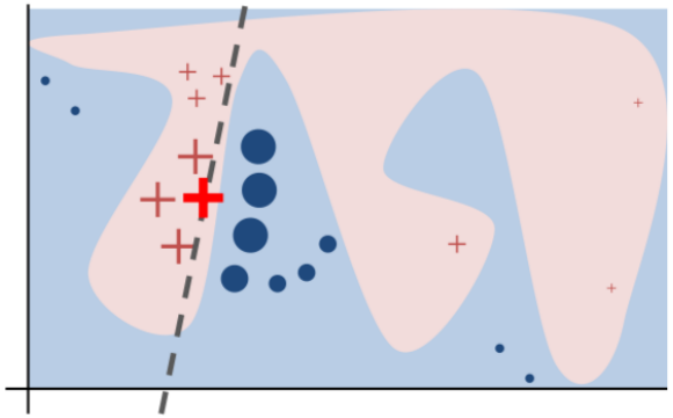
Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro et al., 2016]

- Remember that a sparse linear model is a good explanation model



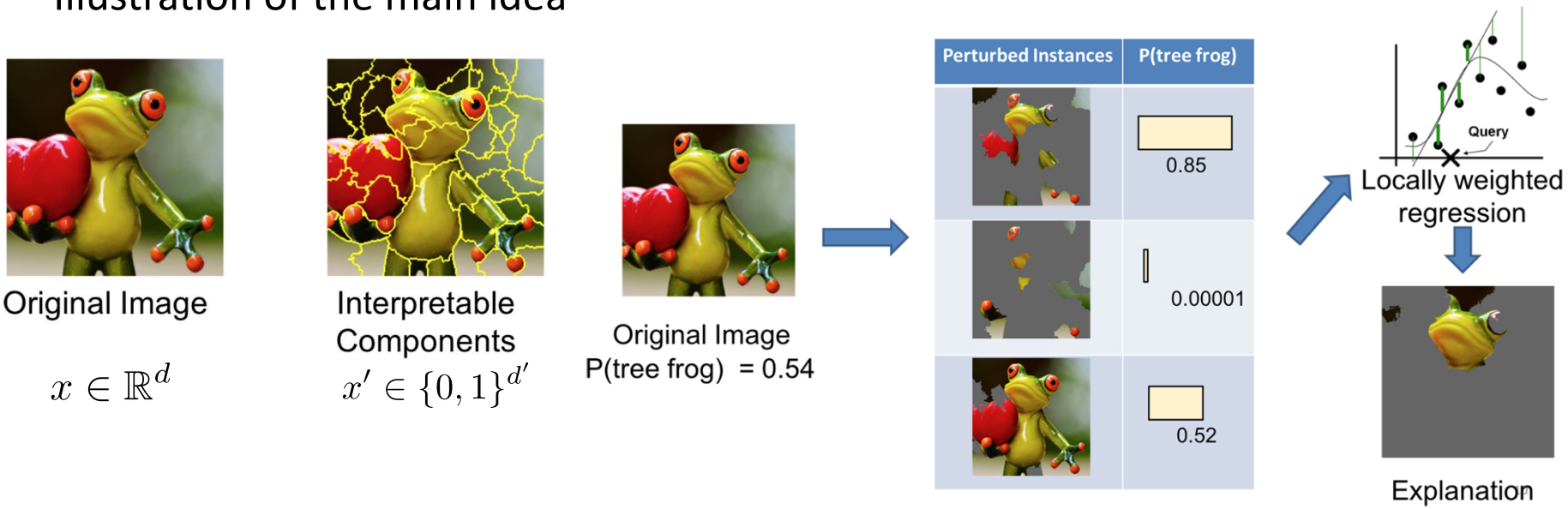
(a) Original Image (b) Explaining *Electric guitar*

- **Idea**: Local linear approximation
  - Explain the entire model is hard, but a single prediction is easier
  - Approximate the model in a local region around the single prediction by a linear classifier



Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro et al., 2016]

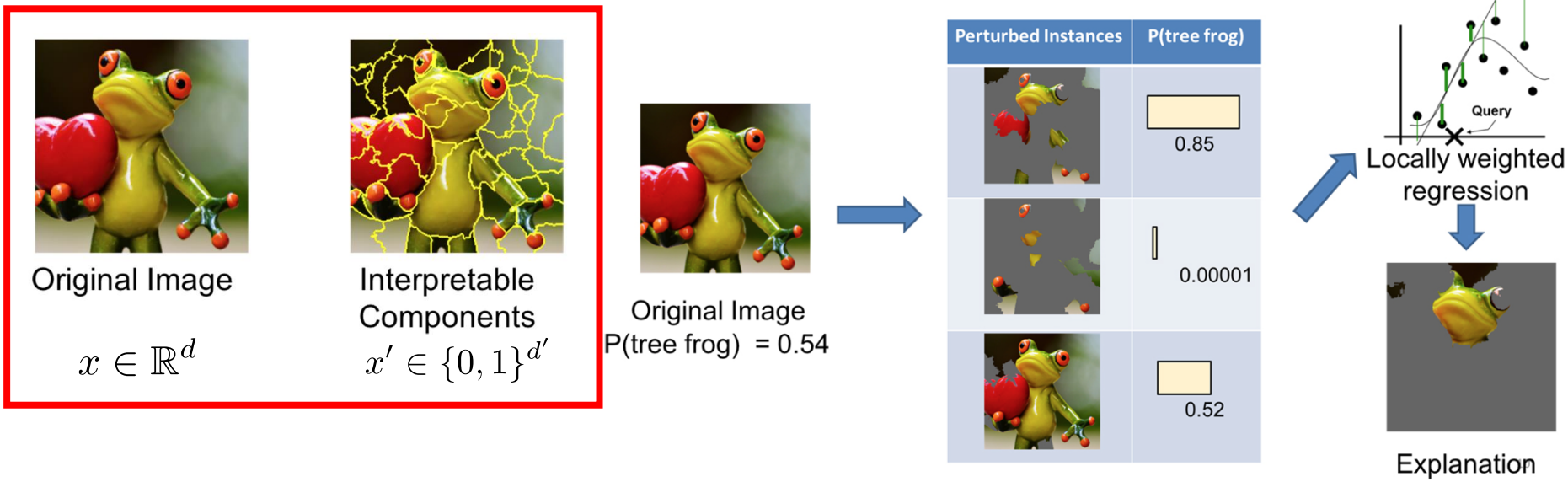
- Illustration of the main idea



- Overall Procedure
  1. Decompose original input to interpretable representation
  2. Model local region around given input by sampling
  3. Approximate original model as a linear classifier

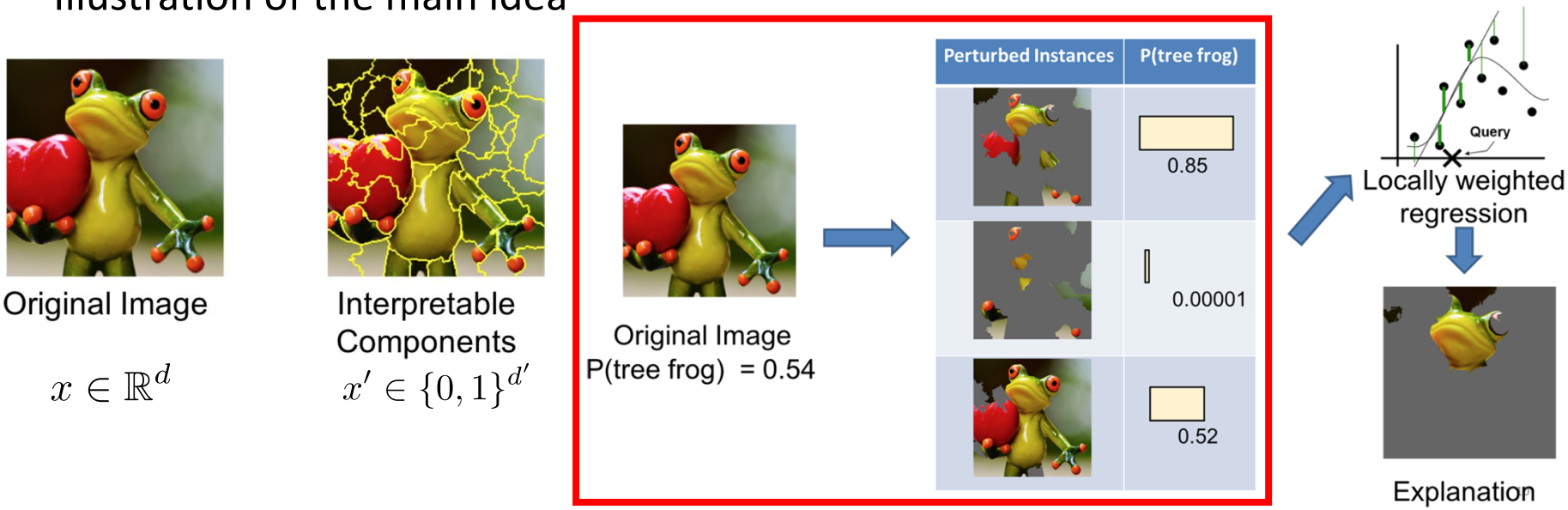
Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro et al., 2016]

- Illustration of the main idea



- **Step 1**: Interpretable representation
  - Understandable to humans
  - For text classification, a binary vector indicating the presence or absence of a word
  - For image classification, a binary vector indicating the presence or absence of a contiguous patch of similar pixels
  - $x \in \mathbb{R}^d$: original representation / $x' \in \{0,1\}^{d'}$: its interpretable representation

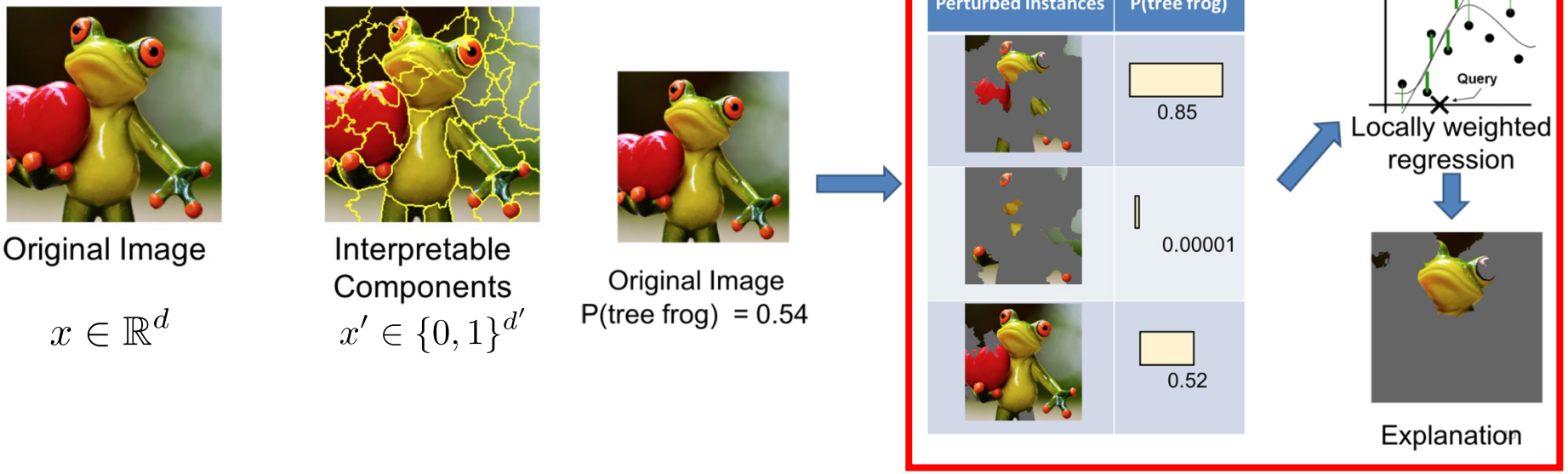
- Illustration of the main idea



- **Step 2**: Model local region around given input
  - Sample instances around x by drawing nonzero elements of $x' \in \{0,1\}^{d'}$ uniformly at random
  - Given a perturbed sample $z' \in \{0,1\}^{d'}$, recover the original representation $z \in \mathbb{R}^d$
  - Compute $f(z)$: the prediction of model for each perturbed output

Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro et al., 2016]

- Illustration of the main idea



- **Step 3**: Approximate original model as a linear classifier
  - Fit a linear classifier $g(z') = w_g \cdot z'$ and use it as an explanation model

$$\mathcal{L}(f, g, \Pi_x) = \sum_{z, z' \in \mathcal{Z}} \Pi_x(z)(f(z) - g(z'))^2$$

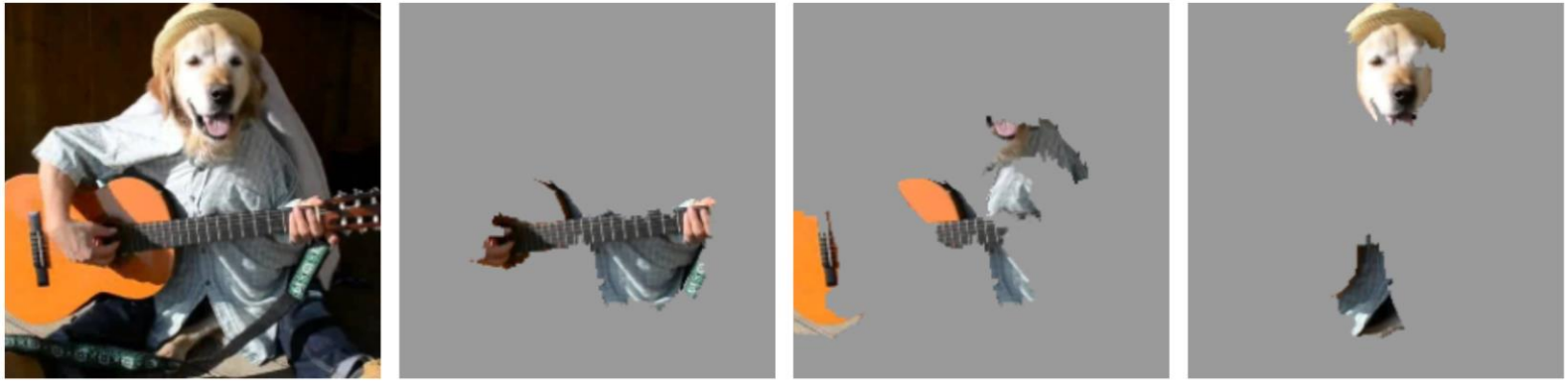
  - $\Pi_x(z)$ defines locality (e.g. $\Pi_x(z) = \exp(-\|x - z\|_2^2/0.1)$)
  - Final objective

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \Pi_x) + \Omega(g)$$

local fidelity — measure of complexity (e.g. L0 norm)

Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro et al., 2016]

- **Results**: Can be applied to any model
  - Top 3 predictions of Inception-v3 for ImageNet dataset



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

  - Random forest prediction for the 20 newsgroups dataset

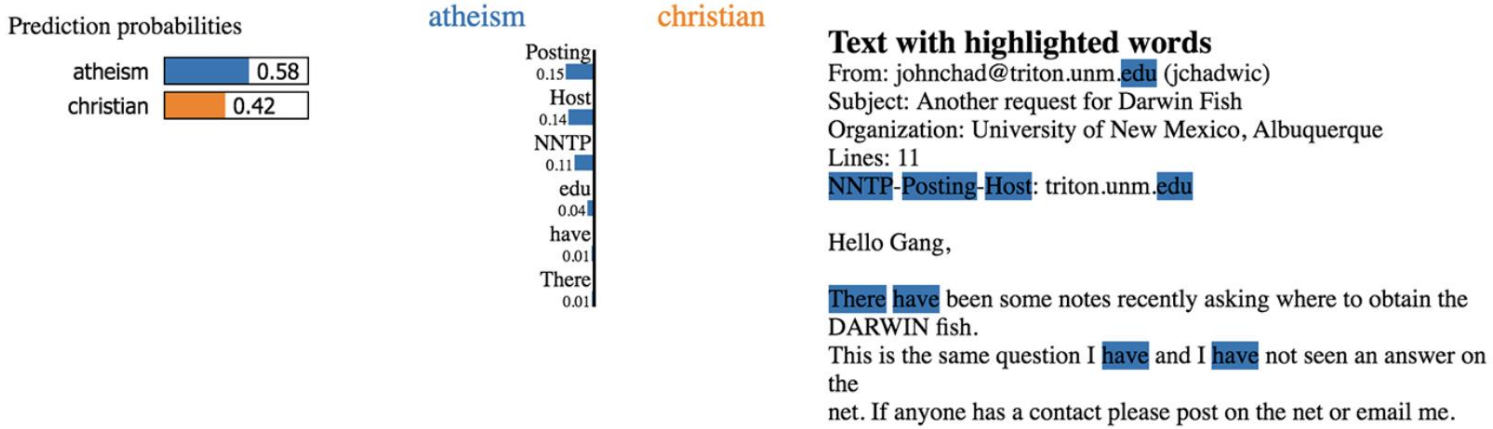


Table of Contents

Saliency Map [Simonyan et al., 2014]

- **Problem**: Perturbation-based methods are too slow
- **Idea:** Use gradient of output with respect to the input as the attribution

- **Goal**: Find the influence on the score $S_c(I_0)$ for given image I_0
  - Consider the linear score model for class c

$$S_c(I) = w_c^\top I + b_c$$

where I : image, w_c, b_c : the weight vector and the bias of the model

  - w_c defines the importance of the corresponding pixels of I for the class c

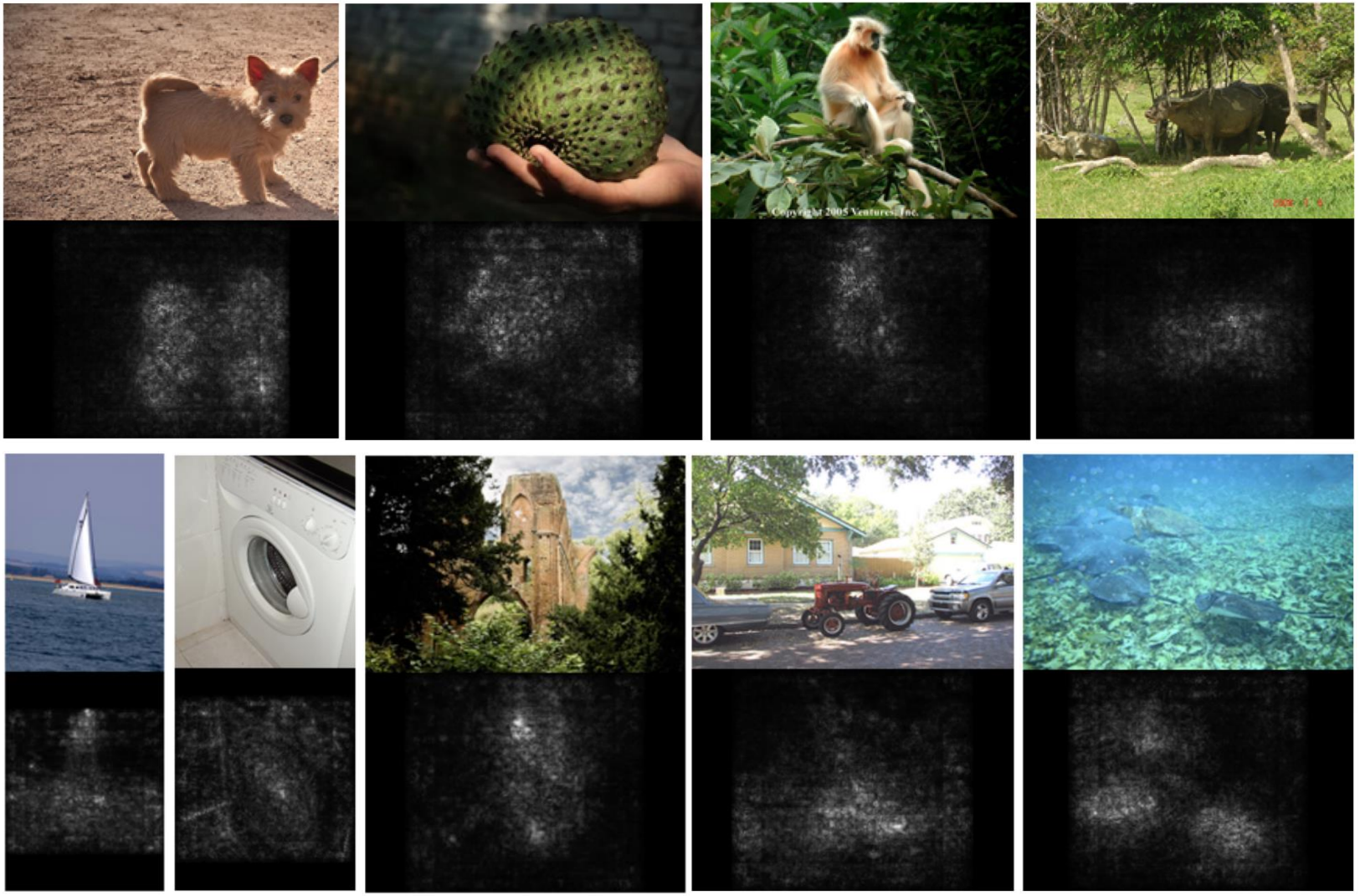
  - In case of non-linear/complex models, approximate $S_c(I)$ by the first-order Taylor expansion

$$S_c(I) \approx w^\top I + b$$

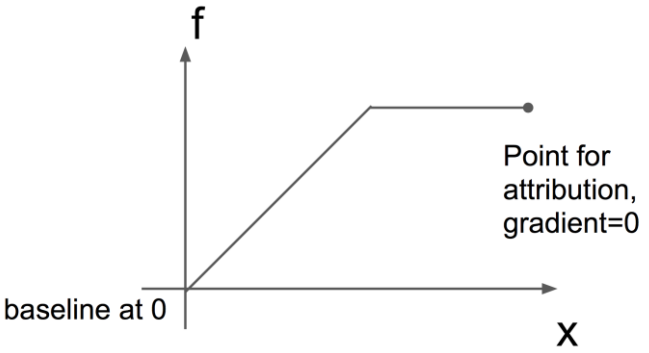
where $w = \left.\frac{\partial S_c}{\partial I}\right|_{I=I_0}$

Saliency Map [Simonyan et al., 2014]

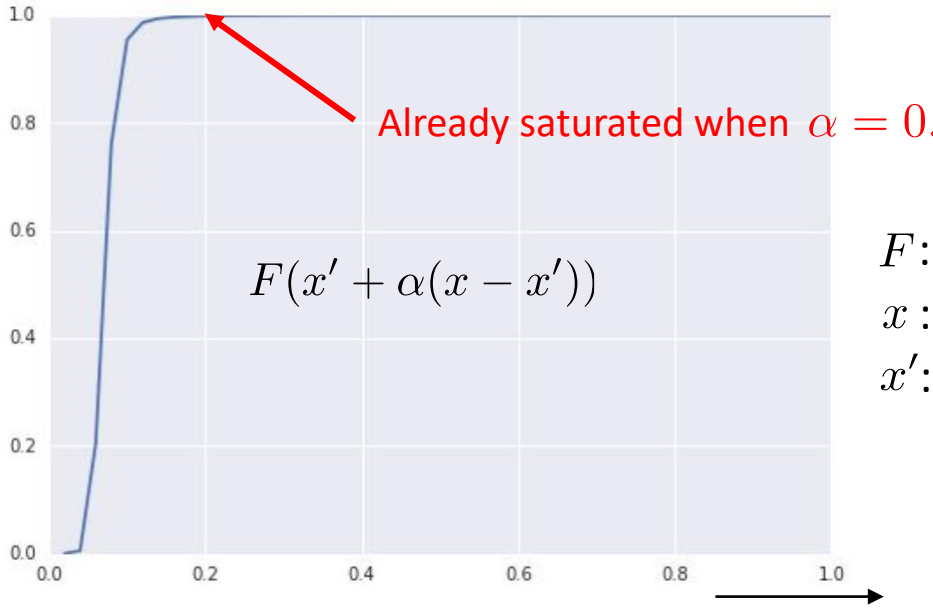
- **Results:** Without any additional annotation, gradient can localize the object



- **Problem**: Prediction score might saturate
  - For high confidence prediction, small perturbation in input does not change the prediction value

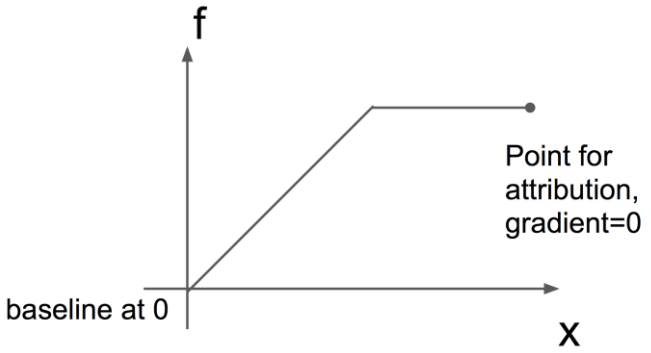


Prediction score

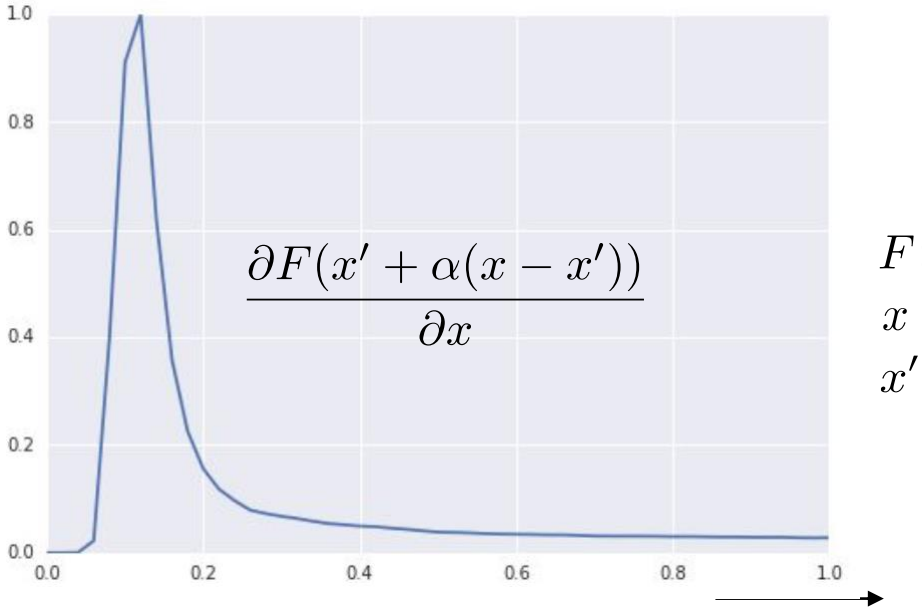


F: prediction score
x : original image
x': baseline image

- **Problem**: Prediction score might saturate
  - For high confidence prediction, small perturbation in input does not change the prediction value



Average pixel gradient (normalized)



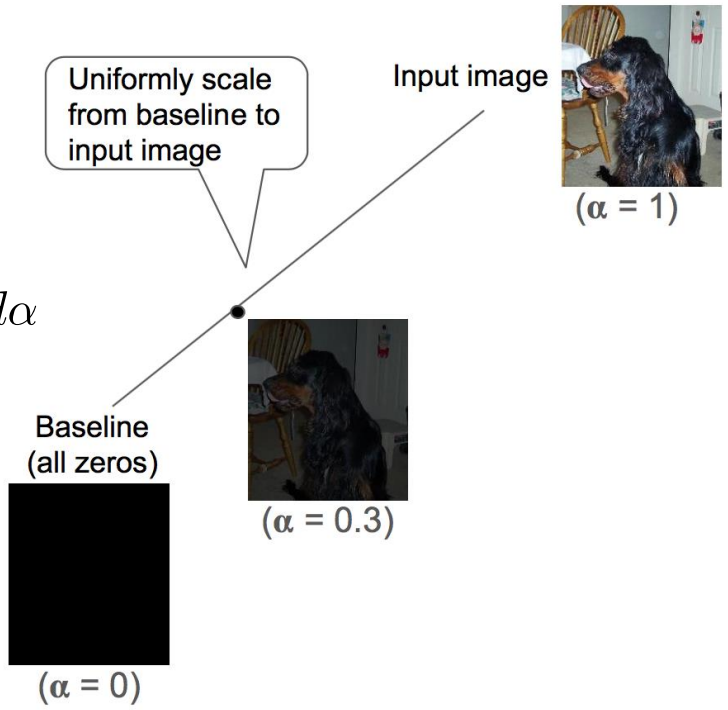
F: prediction score
x : original image
x': baseline image

- **Idea**: Compute all the gradients for images from baseline to actual image

- Construct a sequence of images interpolating from a baseline (black) to the actual image
- Average the gradients across these images

$$\mathrm{IG}_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$

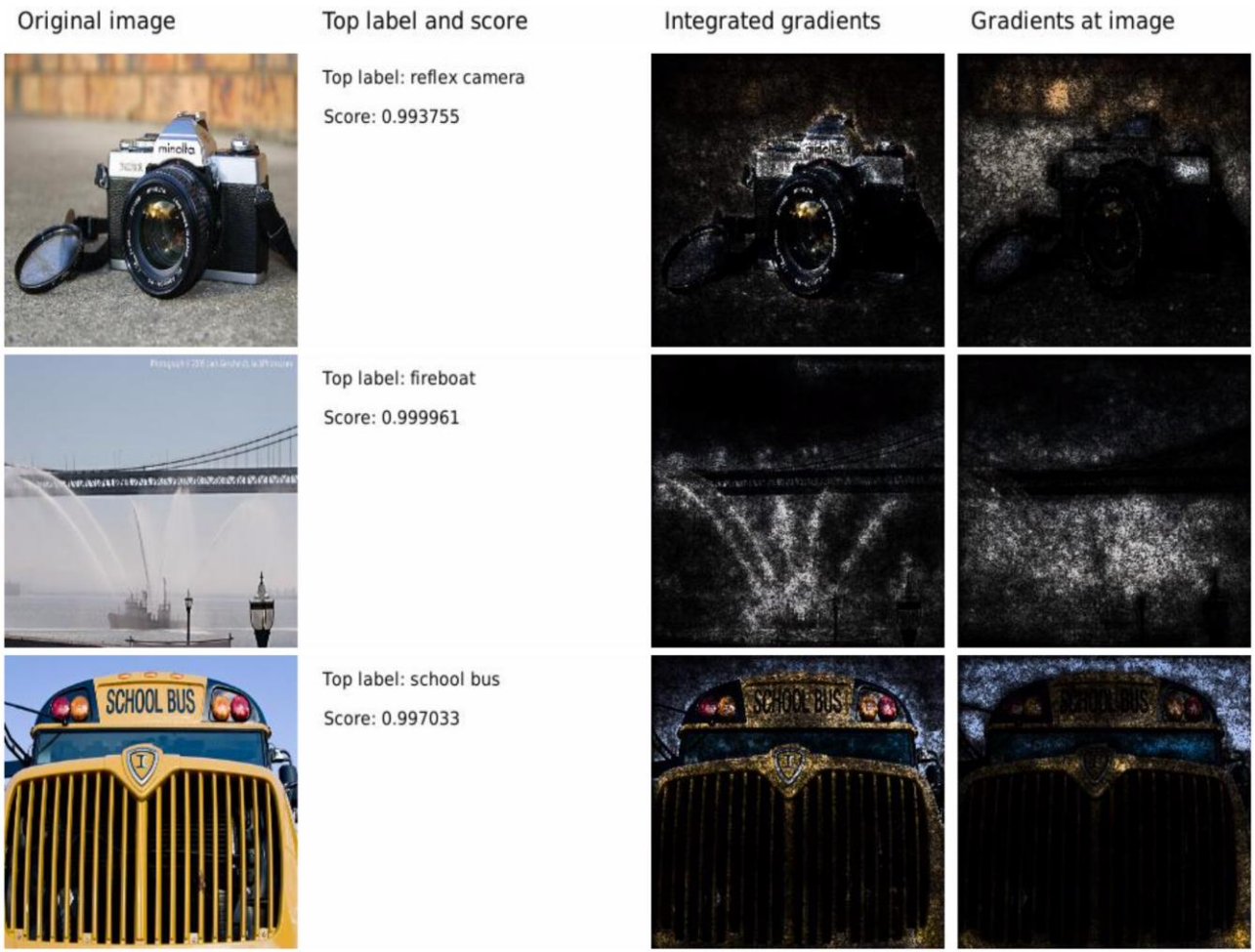
  - F is the prediction function for the label
  - x_i is the intensity of ith pixel
  - $\mathrm{IG}_i(x)$ is the integrated gradient w.r.t. the ith pixel



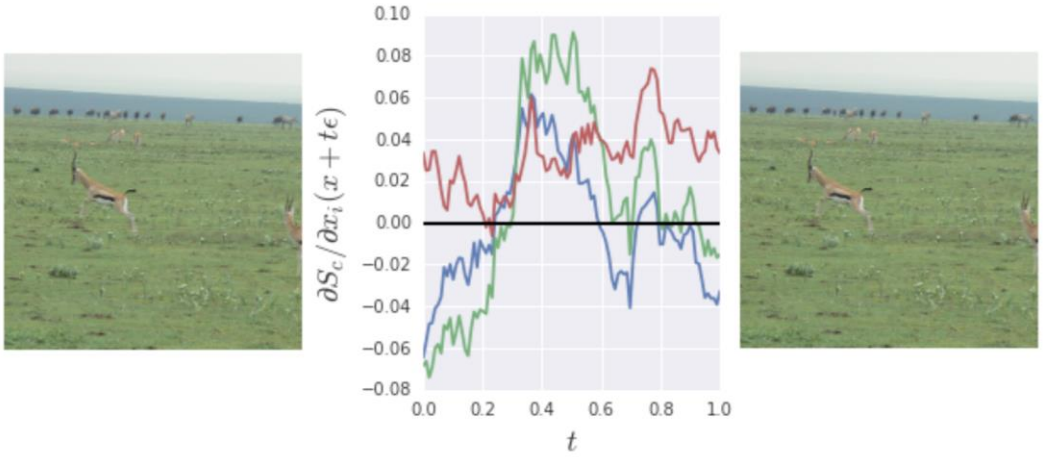
- Properties
  - **Sensitivity**: A variable changes output, then the variable should get an attribution
  - **Insensitivity**: A variable has no effect on the output gets no attribution
  - **Completeness**: $\sum_{i=1}^{n} \mathrm{IG}_i(x) = F(x) - F(x')$

Integrated Gradients [Sundararajan et al., 2017]

- **Results:** For high confidence predictions, IG provide discriminative region



- **Problem**: Gradients strongly fluctuate!
  - Given image x, and an image pixel x_i, plots values of $\max_i \frac{\partial S_c}{\partial x_i}(x + t\epsilon)$ for a short line segment $x + t\epsilon$



  - Even x and $x + \epsilon$ are indistinguishable, the partial derivative rapidly fluctuate

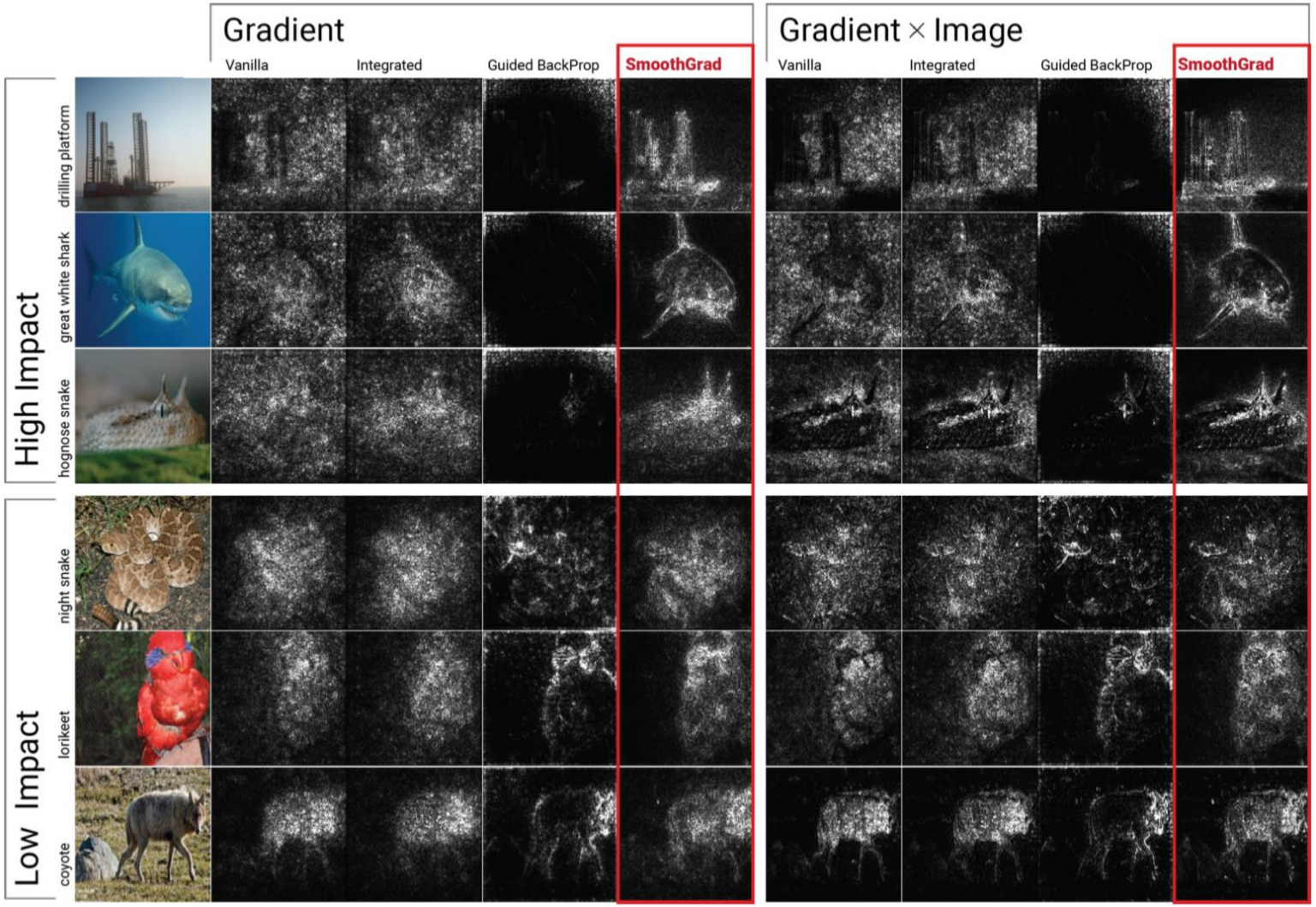
- **Idea**: Use a local average of gradient values

$$\mathrm{SG}(x) = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial S_c}{\partial x}(x + g_i)$$

where noise vectors $g_i \sim \mathcal{N}(0, \sigma^2)$ are drawn i.i.d. from a normal distribution

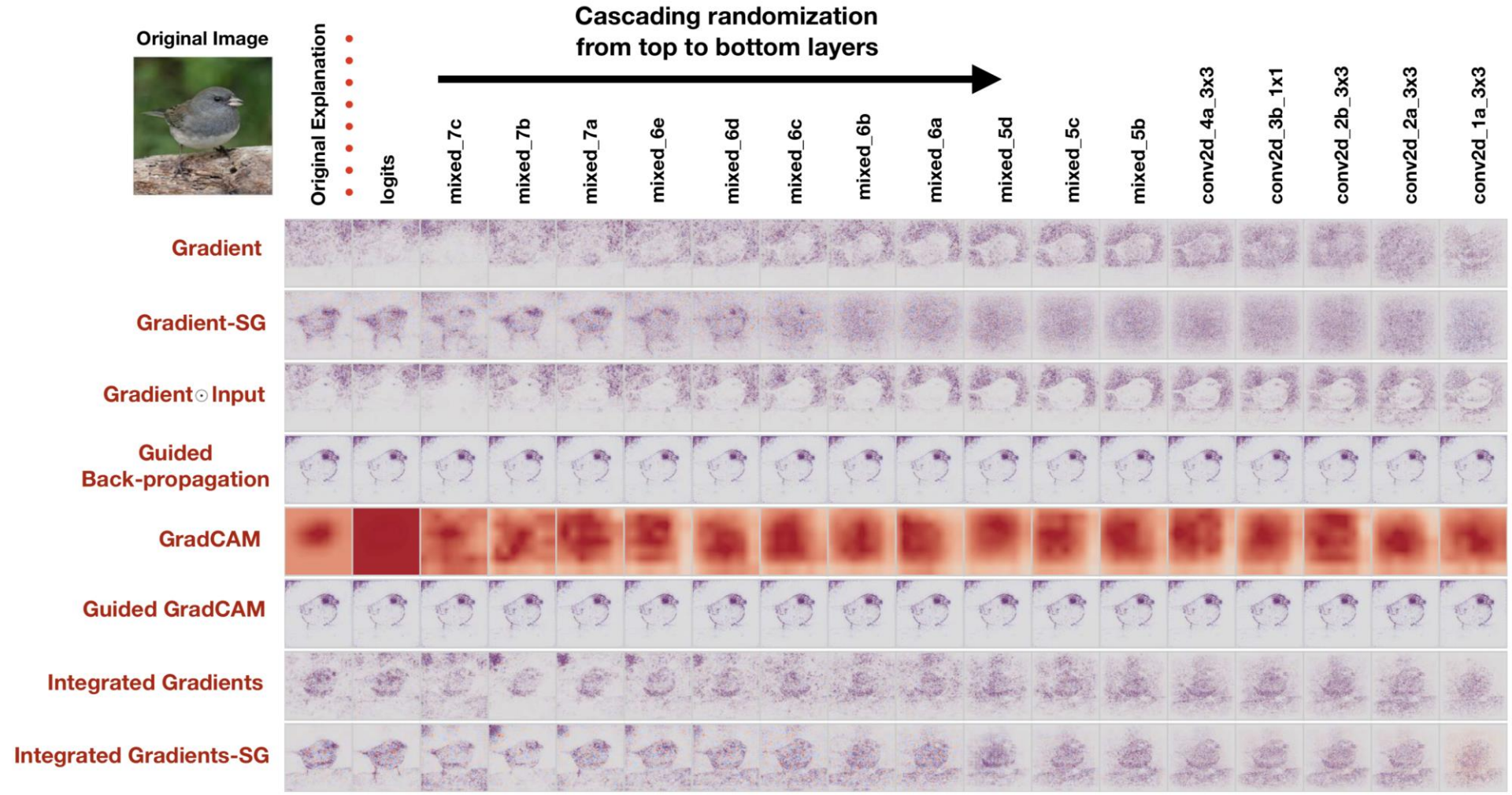
SmoothGrad [Smilkov et al., 2017]

- **Results:** Simple noise-adding method can dramatically improve the quality of saliency map



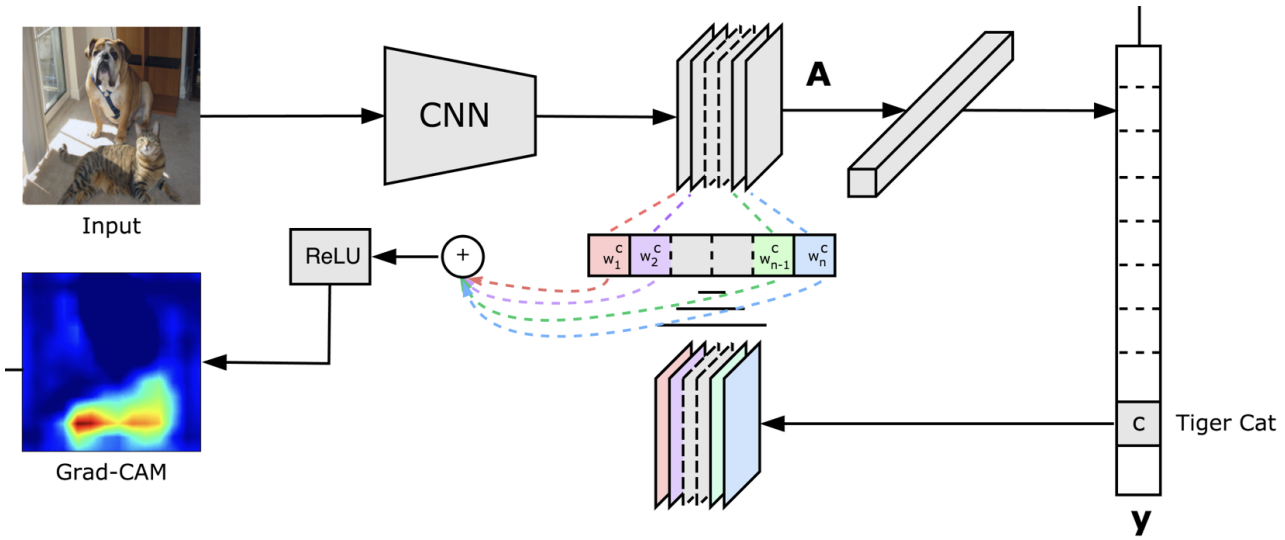
Grad-CAM [Selvararaju et al., 2017]

- **Problem**: Many pixel-level attribution methods insensitive to model parameter [Adebayo et al., 2018]



- **Idea**: Activation-level attribution instead of pixel-level attribution
- Gradient-based extension of CAM [Zhou et al., 2015]
- Can be applied to any CNN based model
  - Image classification, image captioning or visual question answering
- Use GAP of gradients instead of weights after GAP layer
  - y^c: the score for class c, A^k: feature map of the last convolutional layer

$$\alpha_k^c = \sum_{i,j} \frac{\partial y^c}{\partial A_{ij}^k} \qquad L_{\text{Grad-CAM}}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$



Grad-CAM [Selvararaju et al., 2017]

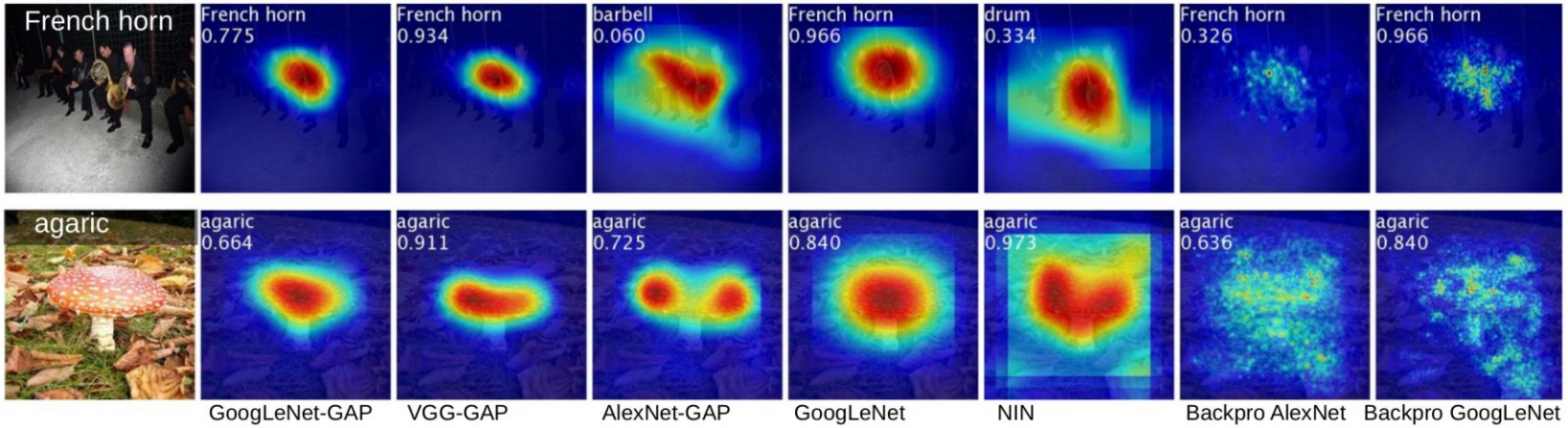
- **Idea**: Activation-level attribution instead of pixel-level attribution
- Gradient-based extension of CAM [Zhou et al., 2015]
- Can be applied to any CNN based model
  - Image classification, image captioning or visual question answering
- Use GAP of gradients instead of weights after GAP layer
  - y^c : the score for class c, A^k : feature map of the last convolutional layer

$$\alpha_k^c = \sum_{i,j} \frac{\partial y^c}{\partial A_{ij}^k} \qquad L_{\text{Grad-CAM}}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

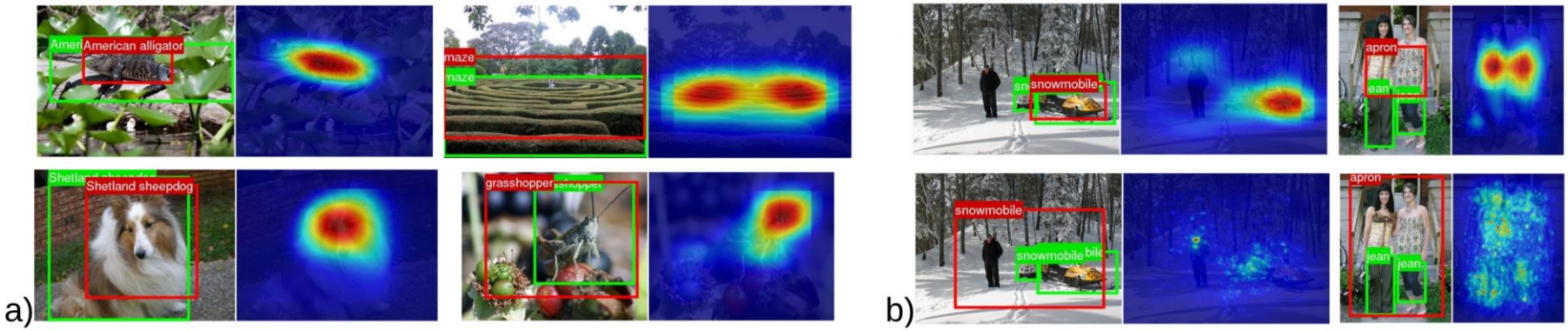
- Typically, the conv activation has low-resolution → low resolution explanation
- Less affected by CNN architecture prior → more sensitive to model parameter

Class Activation Map (CAM) [Zhou et al., 2015]

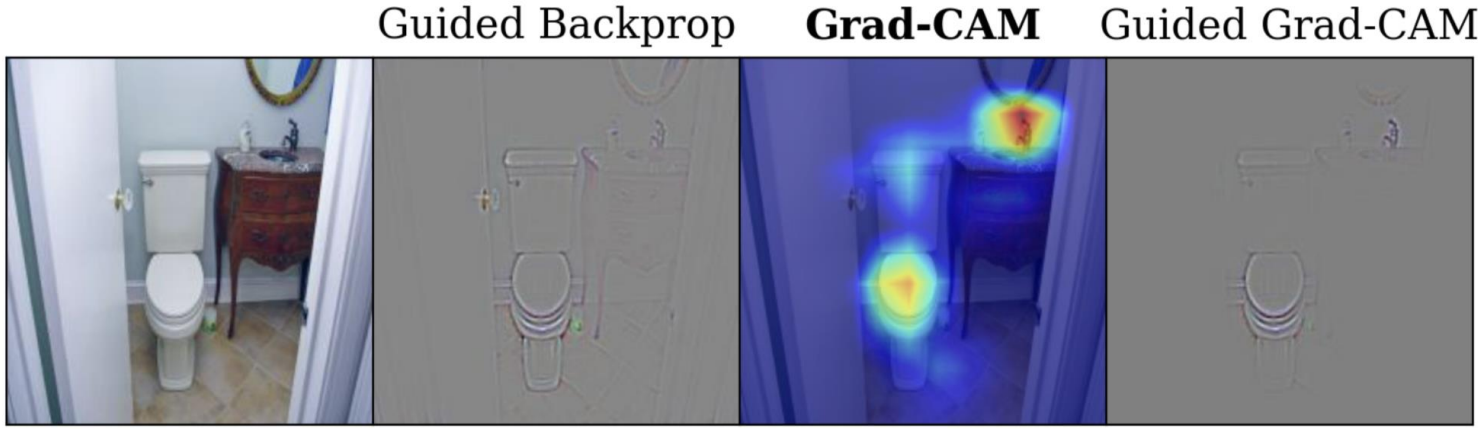
- **Results**
  - CAM vs. Saliency map



  - Examples of localization (green: ground truth / red: predicted)

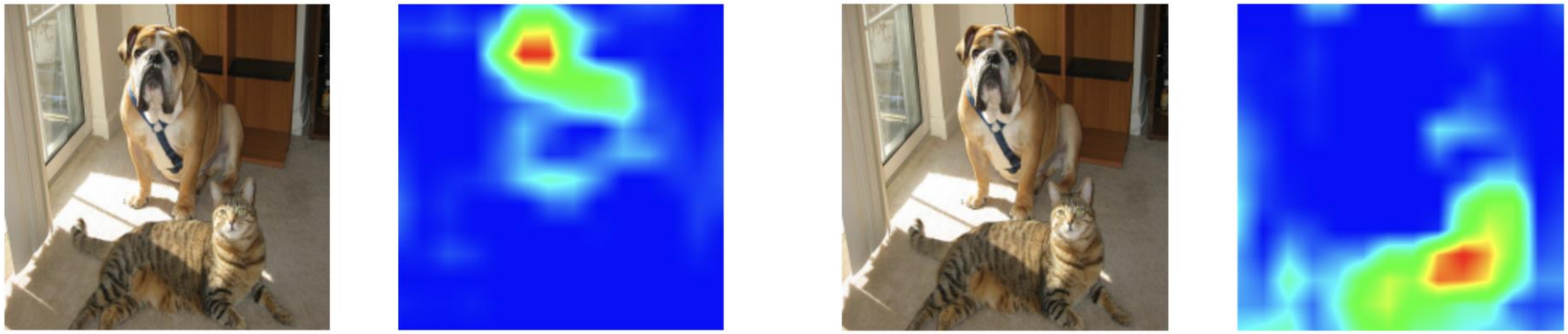


- **Results**: focus on right place without any attention module
  - Visual explanations for captioning



- **Results**: can discriminate different objects
  - Visual explanations for VQA

What animal is in this picture? (left) Answer: dog / (right) Answer: cat



What color is the hydrant? (left) Answer: yellow / (right) Answer: green

Table of Contents

- **Question**: Are hidden units of the trained network align with human concept?

- **Idea**: Make a dataset with *human concepts* as labels (Broden)
  - Gather images from various dataset
  - Contain examples of a broad range of objects, scenes, object parts, textures, and materials in a variety of contexts
  - Most examples are segmented down to the pixel level
  - Total 63,305 pixel-level annotated images, 1,197 visual concepts

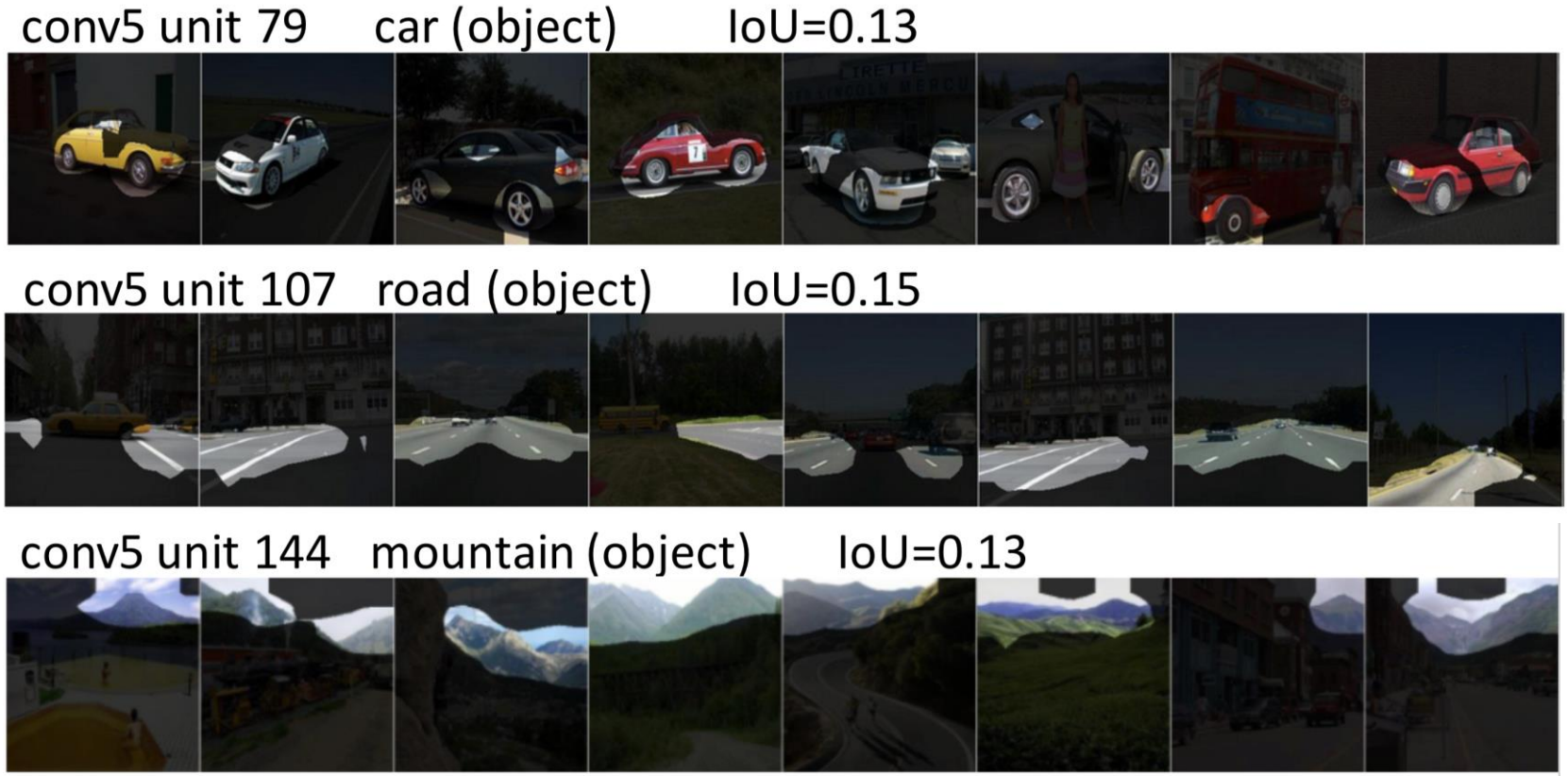
- **Quantifying interpretability** of hidden units
  - For every input image $\mathbf{x}$ in the Broden dataset,
    collect the activation map $A_k(\mathbf{x})$ of every convolutional unit k
  - Define the binary segmentation $M_k(\mathbf{x}) = \mathbf{1}\{S_k(\mathbf{x}) \geq T_k\}$
  - $S_k(\mathbf{x})$: scaled up activation map of $A_k(\mathbf{x})$ (same size as the image)
  - T_k : some threshold value

  - The score of unit k for concept c is reported as a dataset-wide IoU score

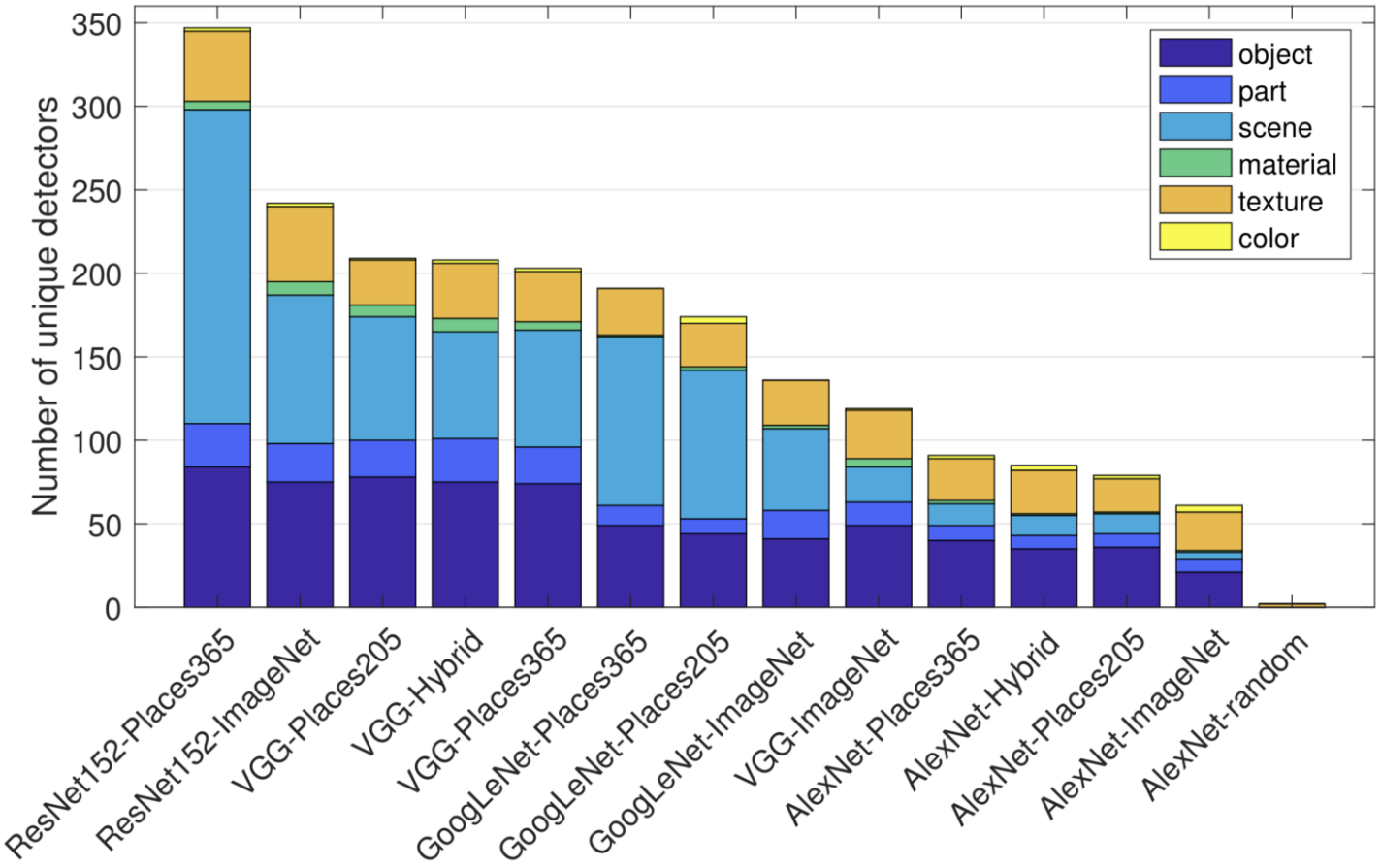
$$IoU_{k,c} = \frac{\sum_{\mathbf{x}} |M_k(\mathbf{x}) \cap L_c(\mathbf{x})|}{\sum_{\mathbf{x}} |M_k(\mathbf{x}) \cup L_c(\mathbf{x})|}$$

  - $L_c(\mathbf{x})$: ground truth mask of image $\mathbf{x}$ for concept c

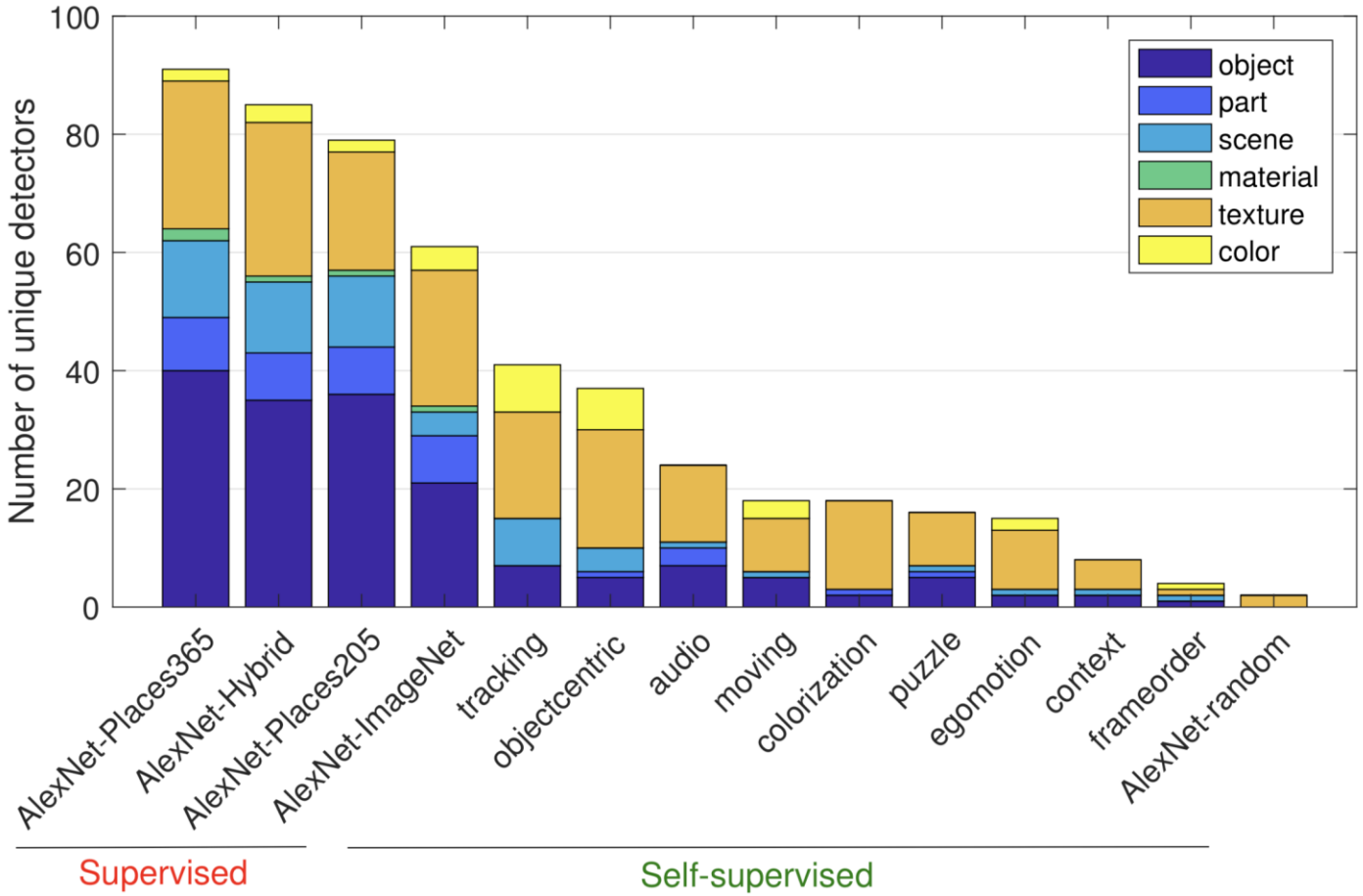
- **Results**: Object detector emerges even when the model trained on scene dataset
  - High-scored (interpretable) convolutional units



- **Results**: Interpretability across different architectures and datasets
  - Deeper architectures appear to allow greater interpretability
  - Scene is composed of multiple objects,
    so it may be beneficial for more object detectors to emerge in CNN



- **Results**: Interpretability across different supervision
  - Self-supervision creates many texture detectors, but relatively few object detectors
  - Colorization trained on colorless images, so that no color detectors



Concept Activation Vector [Kim et al., 2018]

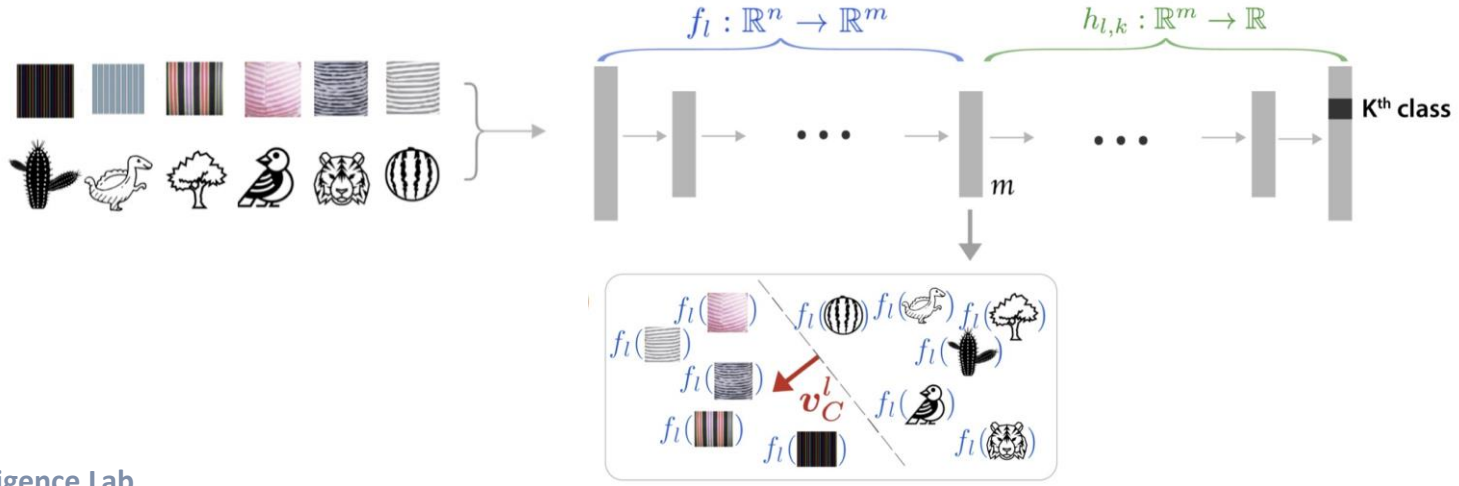
- **Question**: How much certain human concept affected the prediction?

- **Idea**: Define *human concept* as a **vector** in the representation space
  - First, define some concept as a **set of examples**

"**striped**" concept examples

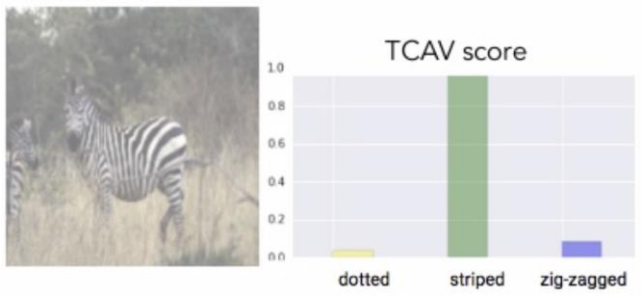
random examples

  - Train a linear classifier to separate concept features and random features
  - **Concept activation vector (CAV)** is the vector orthogonal to the decision boundary



- Quantifying **conceptual sensitivity**
  - *Conceptual sensitivity* of class k to concept C

$$S_{C,k,l}(\mathbf{x}) = \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(\mathbf{x}) + \epsilon \mathbf{v}_C^l) - h_{l,k}(f_l(\mathbf{x}))}{\epsilon}$$
$$= \nabla h_{l,k}(f_l(\mathbf{x})) \cdot \mathbf{v}_C^l$$



  - $h_k(\mathbf{x})$: logit for a data point $\mathbf{x}$ for class k
  - $\mathbf{v}_C^l$: unit concept activation vector for a concept C

$$S_{C,k,l}(\mathbf{x}) = \frac{\partial h_k(\mathbf{x})}{\partial \mathbf{v}_C^l}$$ ← **zebra**-ness ← **striped**-CAV

- Testing with CAV
  - Measure how much specific **concept** is related to certain **class**
  - Fraction of class k inputs whose l layer activation vector is positively influenced by concept C

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

$S_{C,k,l}($  $)$

$S_{C,k,l}($  $)$

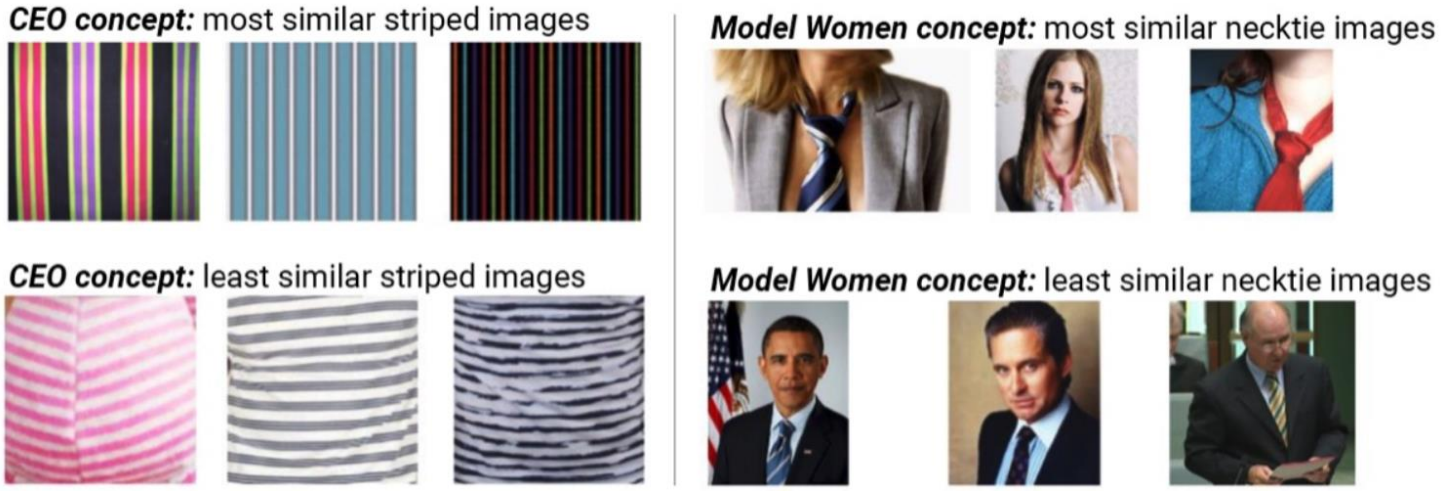
$S_{C,k,l}($  $)$

$S_{C,k,l}($  $)$

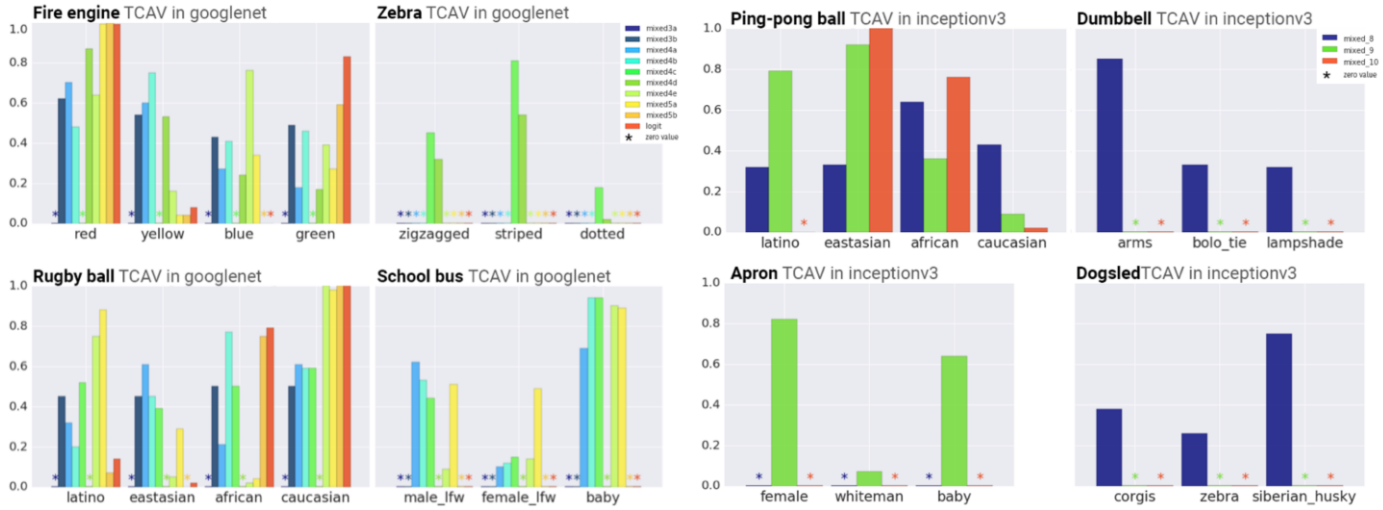
  - Quantifying *global behavior* of the model

Concept Activation Vector [Kim et al., 2018]

- Sorting images with CAVs

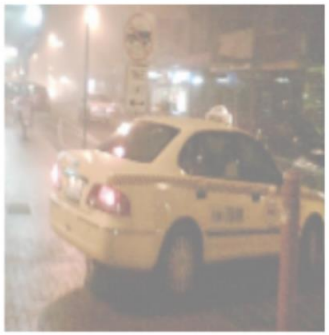


- TCAVs for image classification networks



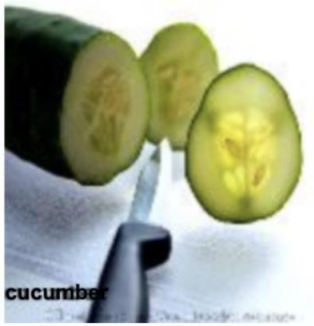
Concept Activation Vector [Kim et al., 2018]

- Simple **sanity check** experiment


cab image

cab image with caption


cucumber image



cucumber with caption

image
+
potentially noisy caption

- Models pay attention to either **image** or **caption** concept for classification
- 4 models trained with different caption noise levels
- Test models with no caption image (**test accuracy = importance of image concept**)

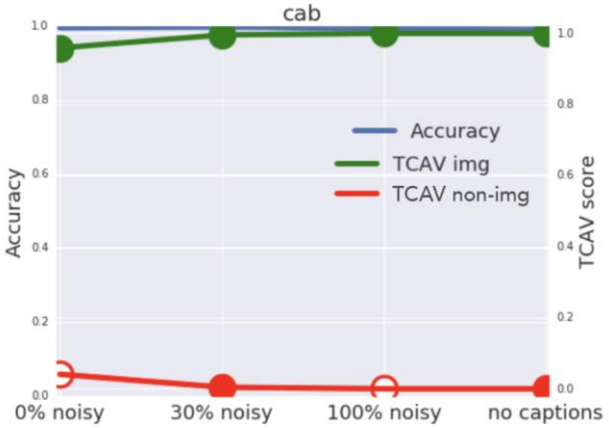
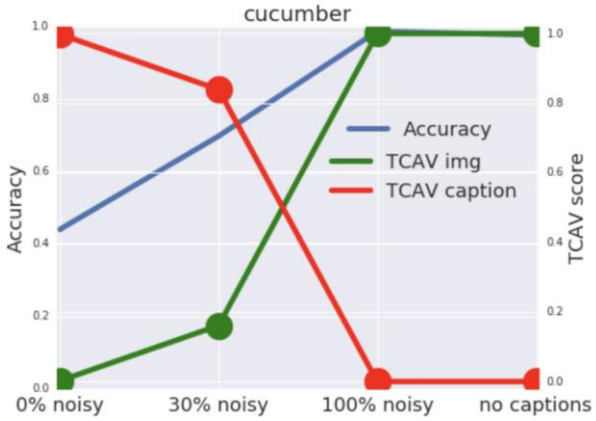


image concept is important

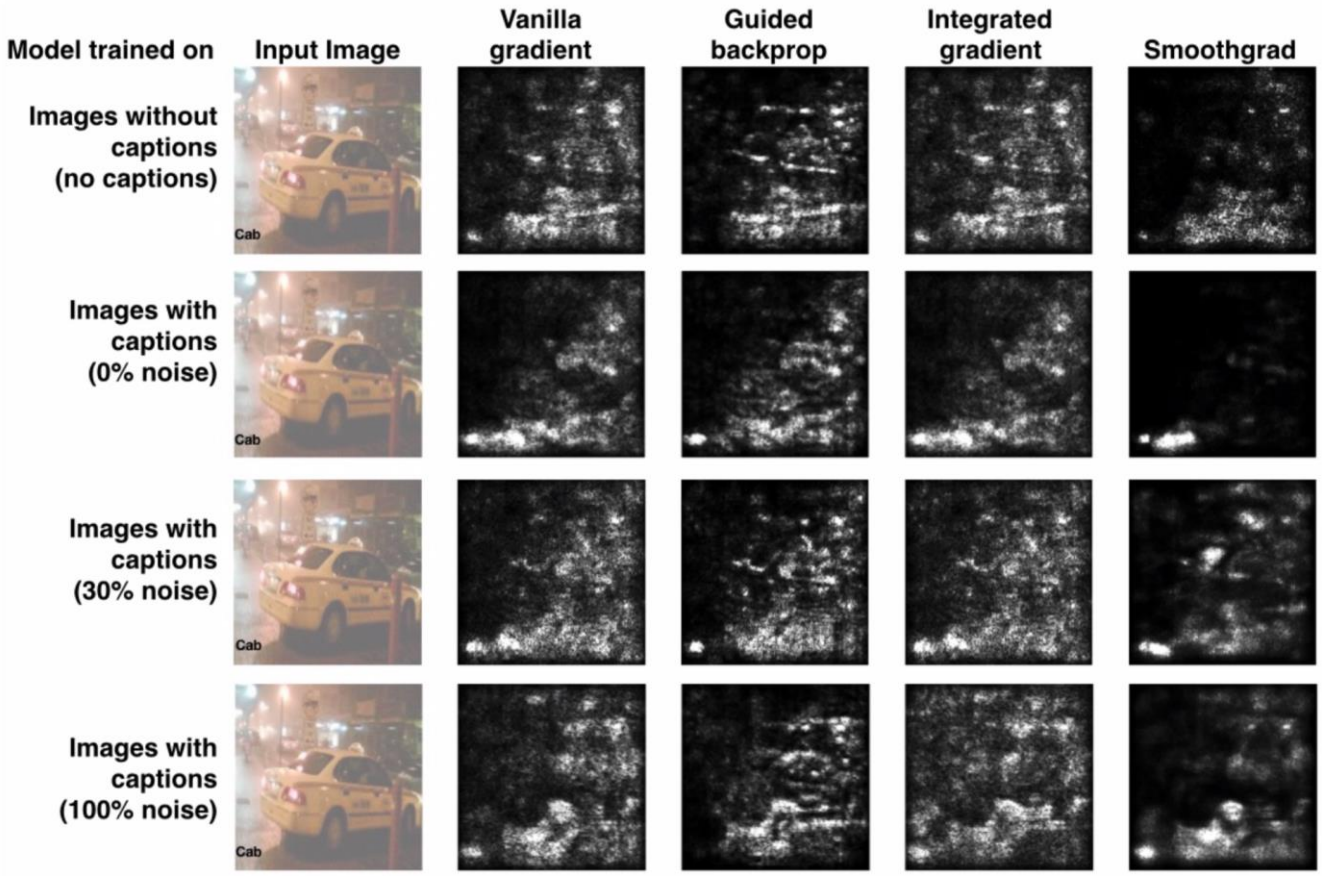


caption concept is important

TCAV score matches with the ground truth well

Concept Activation Vector [Kim et al., 2018]

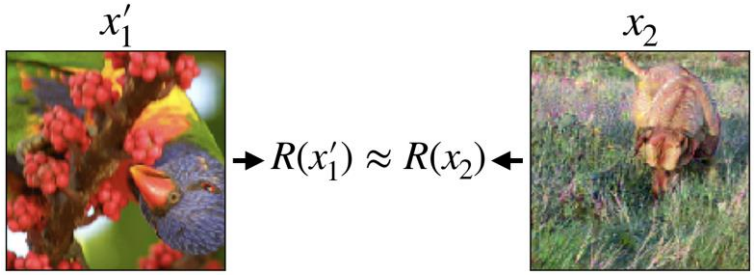
- Simple **sanity check** experiment
  - What about saliency map?



  - None of these model looked at the caption, but saliency map highlights the caption
  - Interpreting using saliency map alone could be **misleading**

- **Question**: How can we obtain human-aligned representation?

- **Problem**: Representation space is **not** *human-aligned*
  - Easy to find two different images with similar representations



$$\|x_1 - x_1'\|_2 \leq \epsilon$$
$$R(x_1') \approx R(x_2)$$

- **Idea**: *Adversarial robustness* as a feature prior
  - Imperceptible changes should not cause large change in prediction

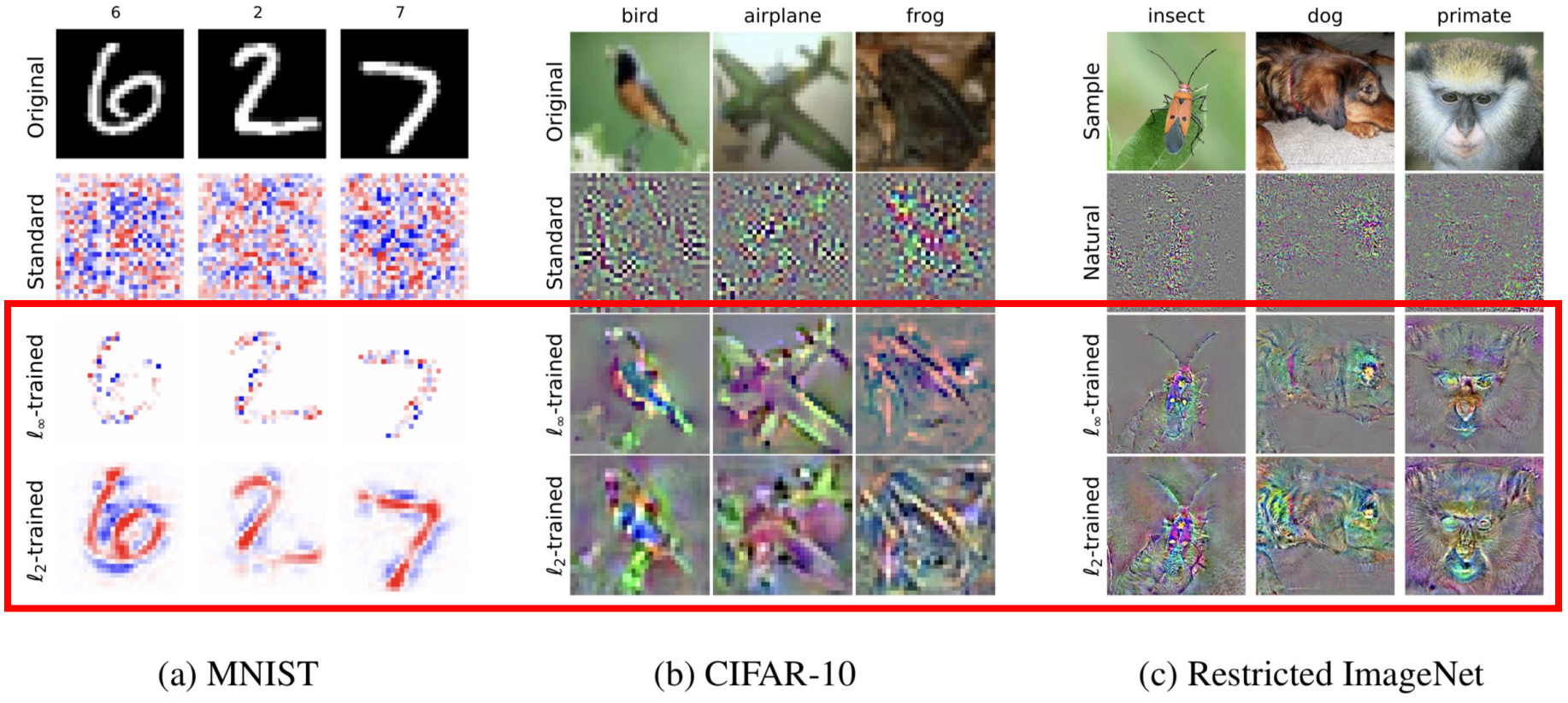
$$\|x - x'\|_2 \leq \epsilon \implies \|f(x) - f(x')\| \leq C \cdot \epsilon$$

  - Note that this is a necessary condition, not a sufficient condition
  - Can enforce this property with adversarial training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\Delta} \mathcal{L}_{\theta}(x+\delta, y) \right]$$

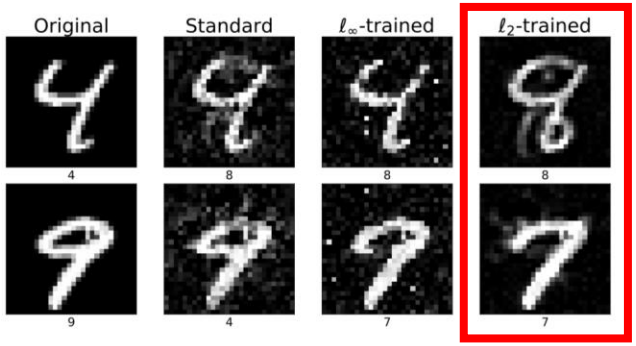
Interpretable Representation via Adversarial Robustness [Engstrom et al., 2019]

- **Results**: Visualizing *loss gradient* with respect to input pixels
  - Gradients are **significantly interpretable** for *adversarially trained* networks

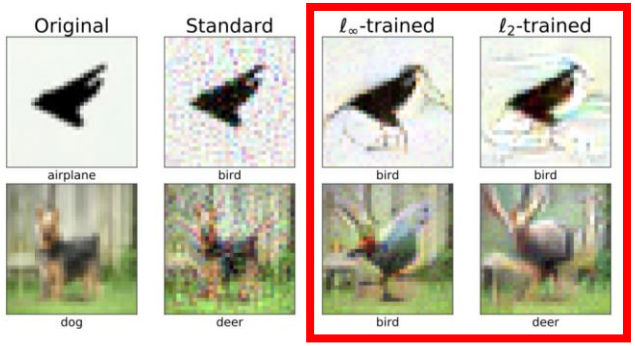


(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

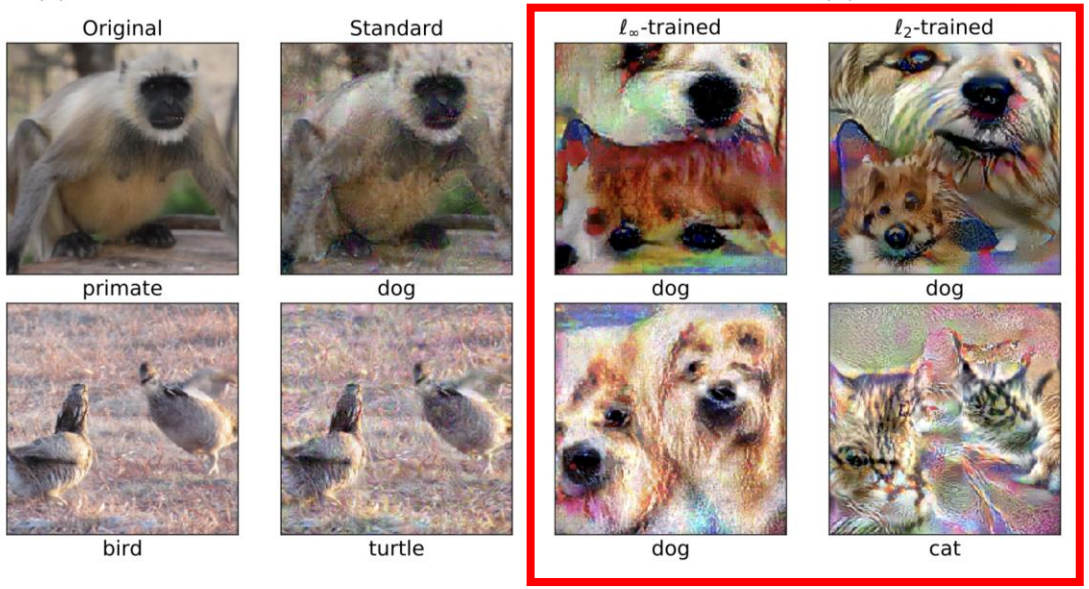
- **Results**: Visualizing large-ϵ *adversarial examples*
  - Adversarial examples for robust models can often be perceived as samples from that class



(a) MNIST



(b) CIFAR-10



(c) Restricted ImageNet

Interpretable Representation via Adversarial Robustness [Engstrom et al., 2019]

- **Results**: Visualizing the *most predictive features*
  - Manipulate input to increase the value of component having the highest weight
  - Provide insight to model's incorrect decision

Summary

- Interpretability is important concept, but there are some obstacles
  - *Hard to define* what exactly interpretability is
  - *Hard to evaluate* interpretability of certain model

- Previous literatures mainly focused on the **feature attribution** problem
  - To find which part of the input is related to the prediction
  - Visual explanation (saliency map / class activation map)

- Recent literatures focus on discover **human-aligned concept**
  - Hidden unit in trained network (network dissection)
  - Vector in the representation space (concept activation vector)
  - Perceptually-aligned representation (L2 adversarial training)

- We are still far from our ultimate goal
  - To understand what's going on inside neural network

References

[Zeiler et al., 2014] Visualizing and Understanding Convolutional Networks. ECCV 2014.
https://arxiv.org/abs/1610.02136

[Zintgraf et al., 2017] Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. ICLR 2017.
https://arxiv.org/abs/1702.04595

[Ribeiro et al., 2016] “Why Should I Trust You?”: Explaining the Predictions of Any Classifier, KDD 2016.
https://arxiv.org/abs/1602.04938

[Simonyan et al., 2014] Deep Inside Convolutional Networks: Visualising Image Classificiation ... ICLR Workshop 2014.
https://arxiv.org/abs/1312.6034

[Sundararajan et al., 2017] Axiomatic Attribution for Deep Networks. ICML 2017.
https://arxiv.org/abs/1703.01365

[Smilkov et al., 2017] SmoothGrad: removing noise by adding noise.
https://arxiv.org/abs/1706.03825

[Selvararaju et al., 2017] Grad-CAM: Visual Explanations from Deep Networks via Gradient-based ... ICCV 2017.
https://arxiv.org/abs/1610.02391

[Zhou et al., 2015] Learning Deep Features for Discriminative Localization. ICCV 2015.
https://arxiv.org/abs/1512.04150

[Adebayo et al., 2017] Sanity Checks for Saliency Maps. NIPS 2018.
https://arxiv.org/abs/1810.03292

[Bau et al., 2017] Network Dissection: Quantifying Interpretability of Deep Visual Representations. CVPR 2017.
https://arxiv.org/abs/1704.05796

References

[Kim et al., 2018] Interpretability Beyond Feature Attribution: Quantitative Testing with Concept … ICML 2018.
https://arxiv.org/abs/1711.11279

[Tsipras et al., 2019] Robustness May Be at Odds with Accuracy. ICLR 2019.
https://arxiv.org/abs/1805.12152

[Engstrom et al., 2019] Learning Perceptually-Aligned Representations via Adversarial Robustness. arXiv preprint.
https://arxiv.org/abs/1906.00945