Computer Vision; Image Classification; Visualizing & Understanding

Boulder

YouTube Playlist

Maziar Raissi

Assistant Professor

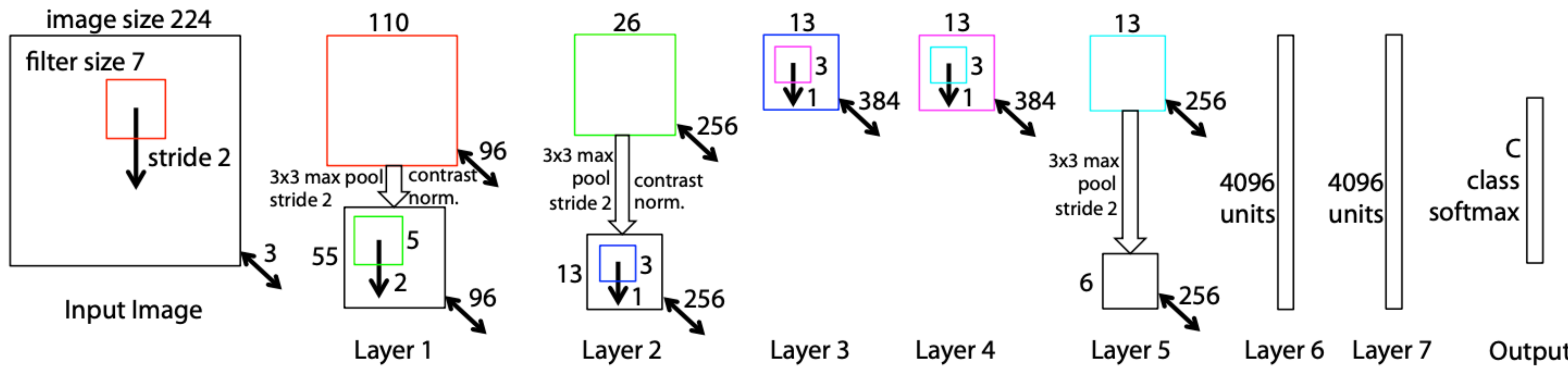
Department of Applied Mathematics

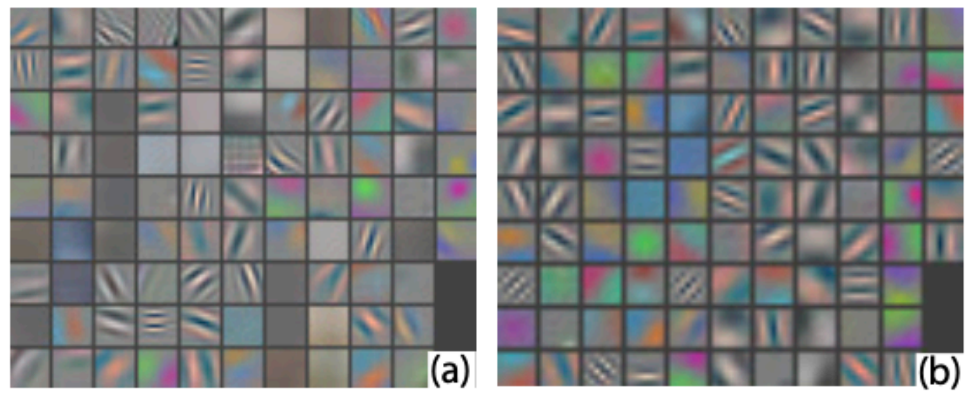
University of Colorado Boulder

maziar.raissi@colorado.edu

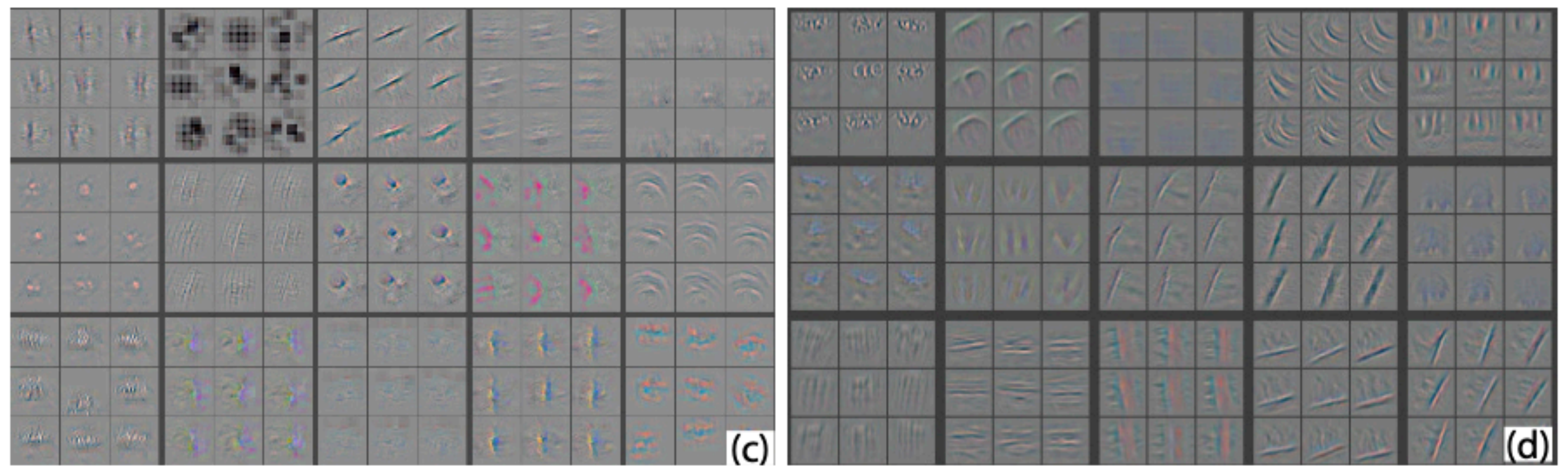
Visualizing and Understanding Convolutional Networks

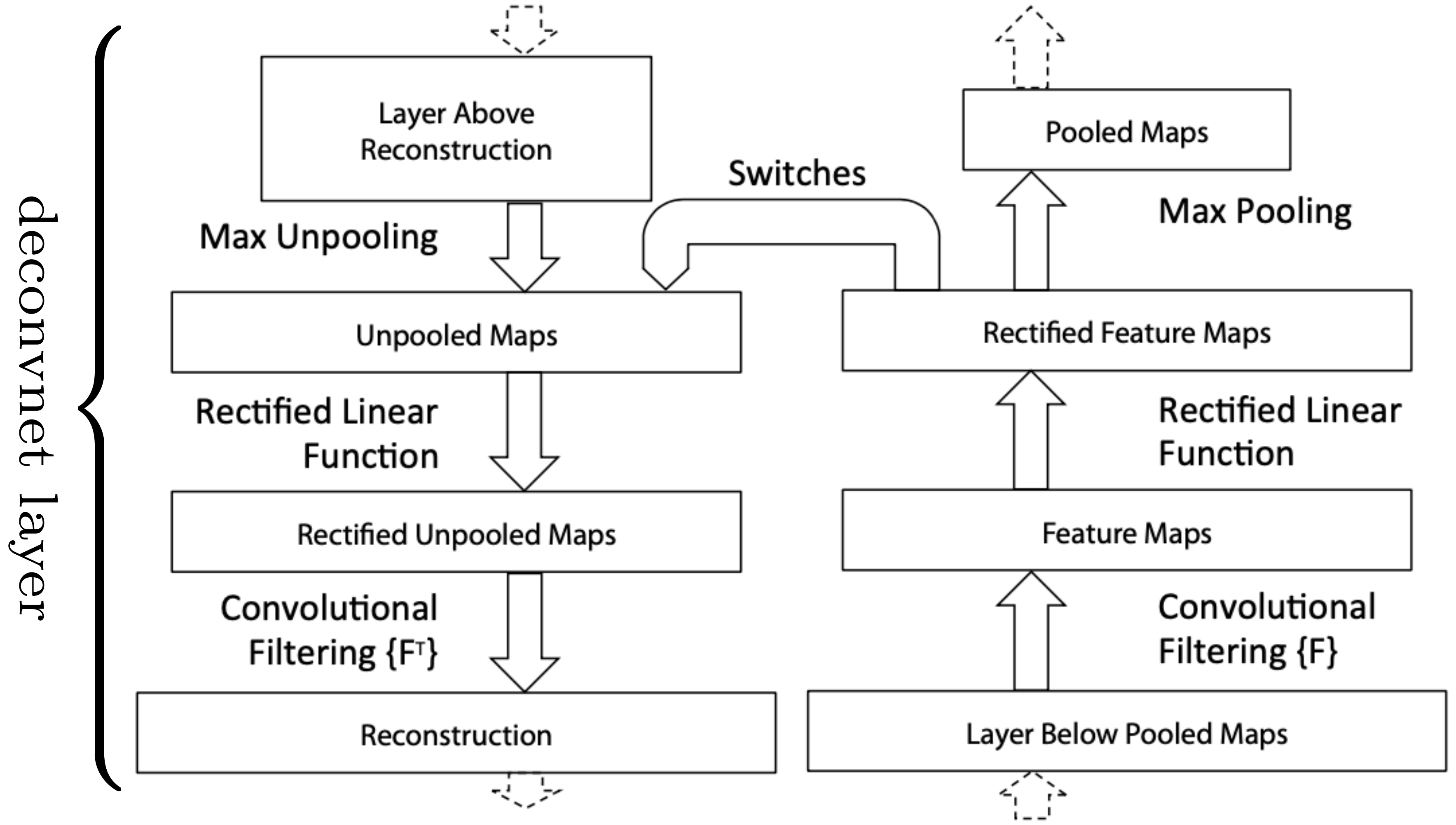
[YouTube Video](#)

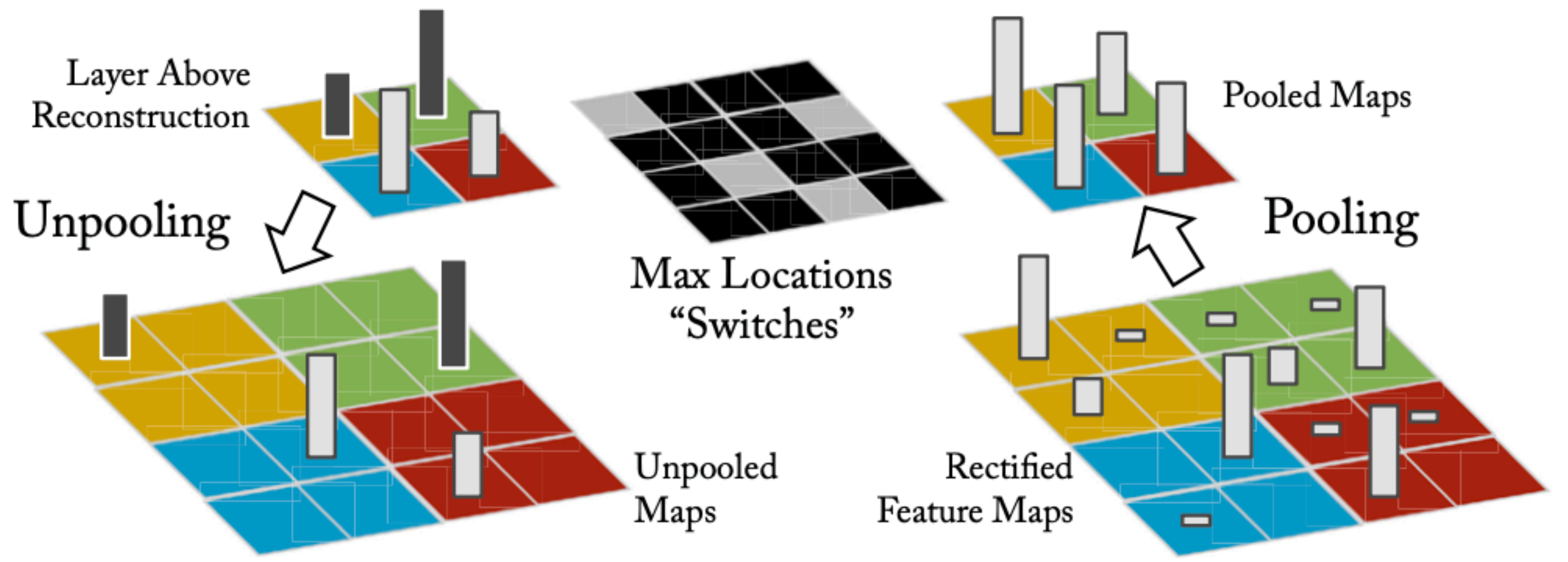




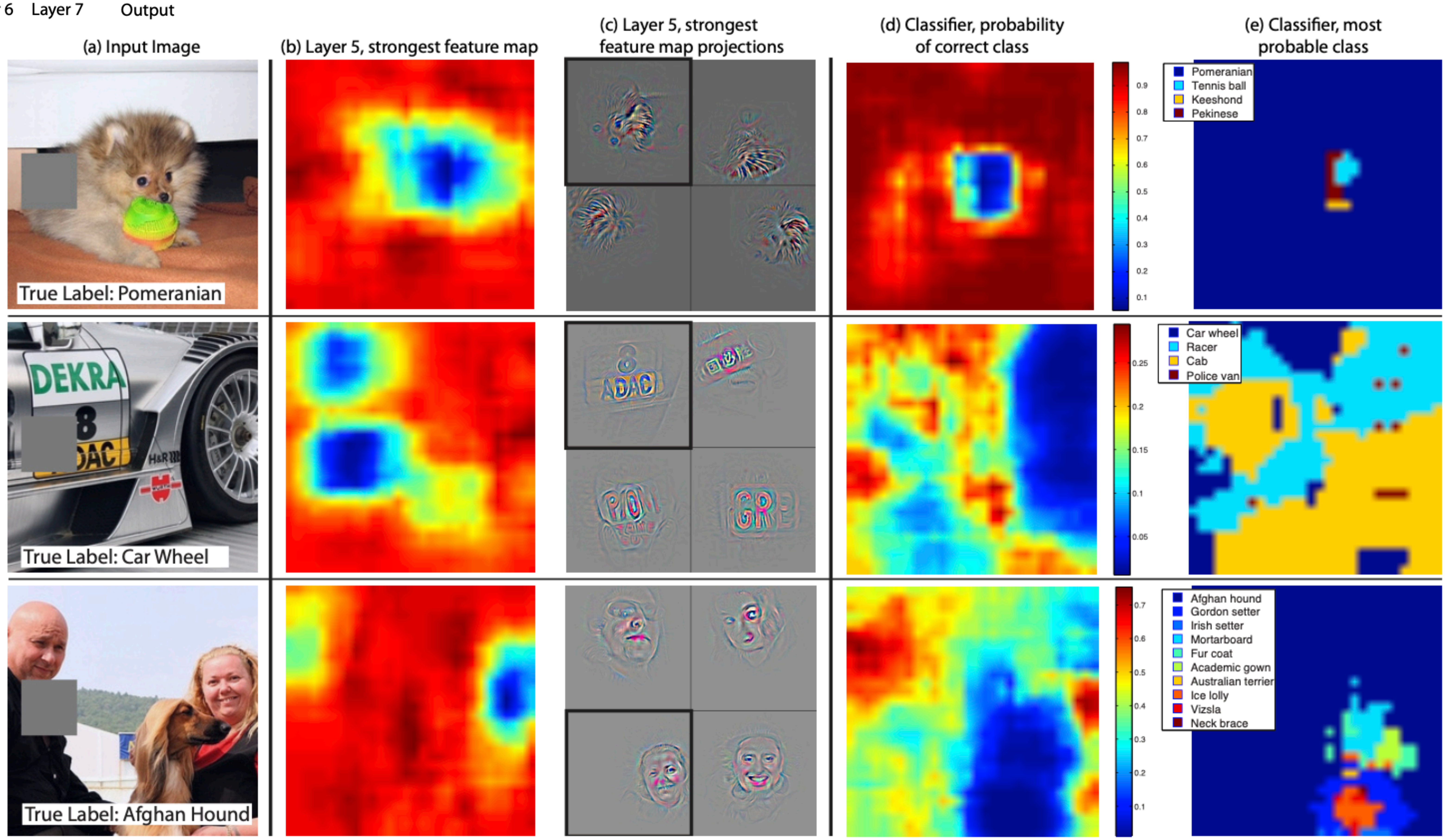
smaller stride (2 vs 4) and filter size (7x7 vs 11x11)





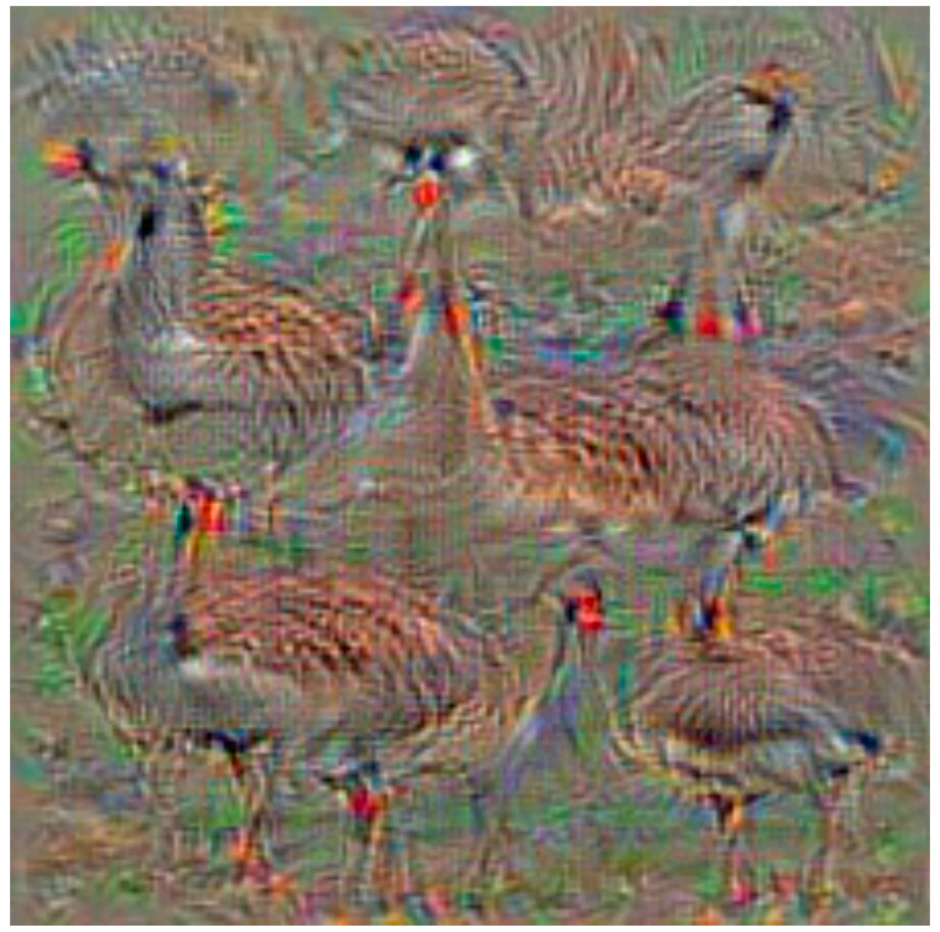


Filtering: Flipping each filter vertically and horizontally

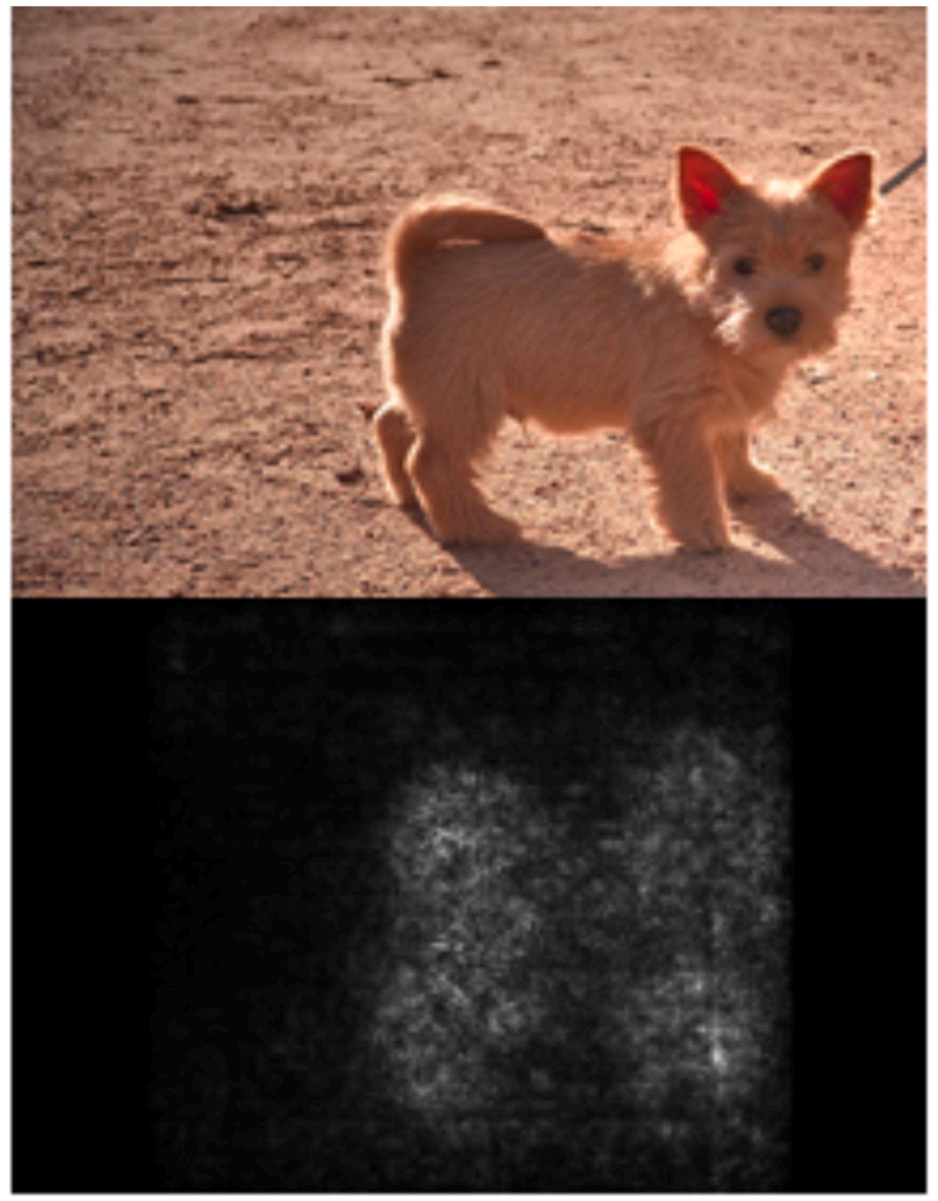


Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps



goose



Class Model Visualization

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

$S_c(I) \rightarrow$ score of class c for image I

$$P_c(I) = \frac{\exp S_c(I)}{\sum_{c'} \exp S_{c'}(I)} \rightarrow \text{probability of class } c$$

$\lambda \rightarrow$ regularization parameter

Image-Specific Class Saliency Visualization

$I_0 \rightarrow$ image

$c \rightarrow$ class

$S_c(I) \approx w_c^T I + b_c$ for I in the neighborhood of I_0

$$w_c = \left.\frac{\partial S_c(I)}{\partial I}\right|_{I=I_0}$$

magnitude of elements of w_c defines the importance of the corresponding pixels of I for the class c

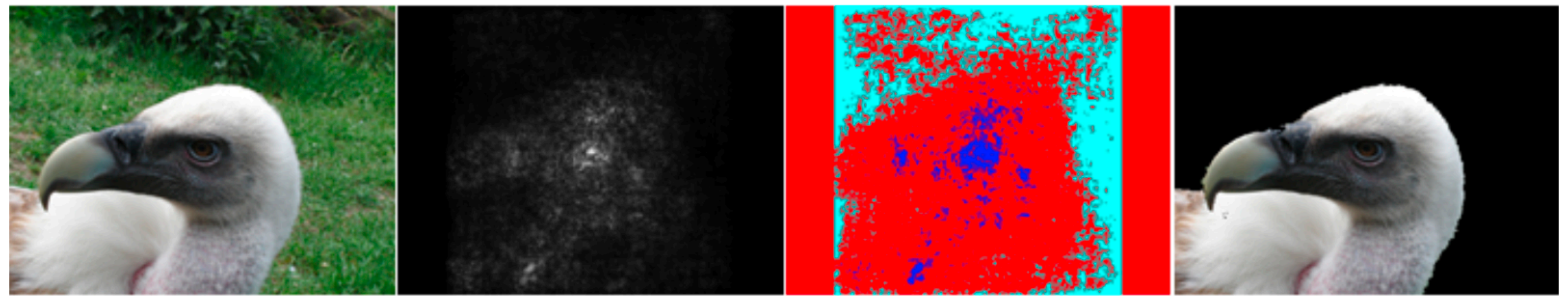
Class Saliency Extraction

$I_0 \in \mathbb{R}^{H \times W \times K} \implies w_c \in \mathbb{R}^{H \times W \times K}$

$M_{ij} := \max_k |w^c_{ijk}| \implies M \in \mathbb{R}^{H \times W}$

Weakly Supervised Object Localization

blue - foreground color model; cyan - background color model; red - not used for color model estimation



Relation to Deconvolution Networks

$X_n \rightarrow$ n-th layer input

$f \rightarrow$ neuron activity to be visualized

$X_{n+1} = X_n * K_n \rightarrow$ convolutional later

$$\frac{\partial f}{\partial X_n} = \frac{\partial f}{\partial X_{n+1}} * \widehat{K}_n$$

$\widehat{K}_n \rightarrow$ flipped version of the convolutional kernel K_n

$R_n \rightarrow$ n-th layer reconstruction in a DeconvNet

$R_n = R_{n+1} * \widehat{K}_n$

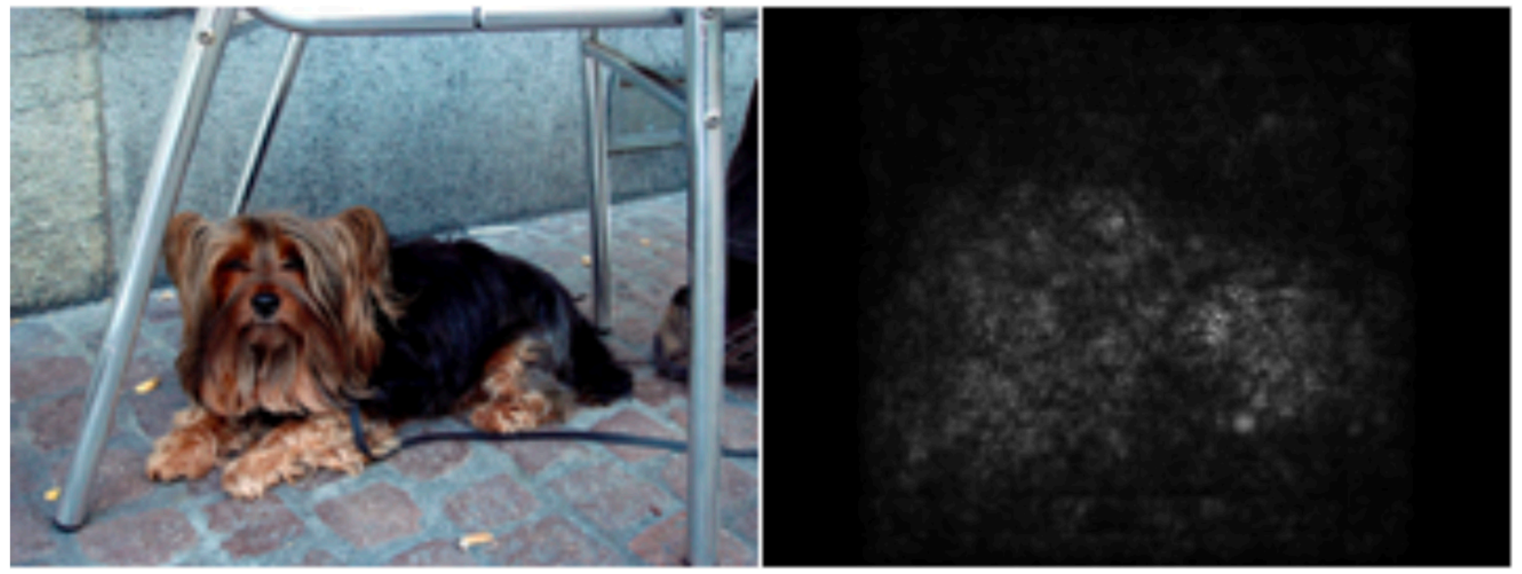
$X_{n+1} = \max(X_n, 0) \rightarrow$ ReLU

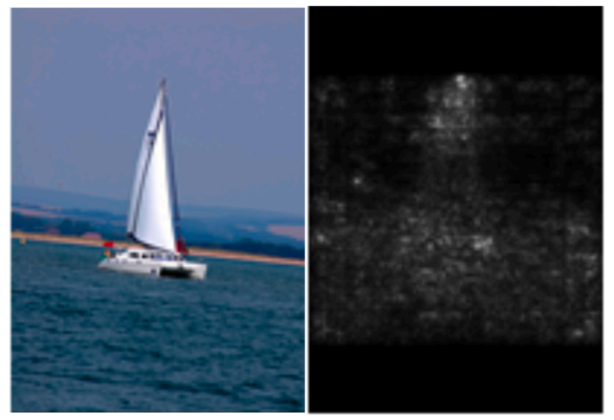
$$\frac{\partial f}{\partial X_n} = \frac{\partial f}{\partial X_{n+1}} \mathbb{1}(X_n > 0)$$

$R_n = R_{n+1} \mathbb{1}(R_{n+1} > 0) \rightarrow$ slightly different from above

$$X_{n+1}(p) = \max_{q \in \Omega(p)} X_n(q) \rightarrow \text{maxpooling}$$

$$\frac{\partial f}{\partial X_n(s)} = \frac{\partial f}{\partial X_{n+1}(p)} \mathbb{1}(s = \arg\max_{q \in \Omega(p)} X_n(q)) \rightarrow \text{switches}$$





Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." *arXiv preprint arXiv:1312.6034* (2013).

Striving for Simplicity: The All Convolutional Net

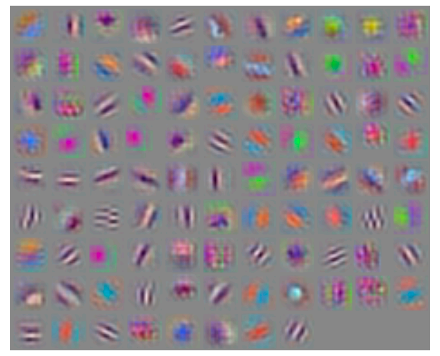
$f \in \mathbb{R}^{H\times W\times N}$ → feature maps produced by some layer of CNN

$$s_{i,j,u}(f) = \left(\sum_{h=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{w=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} |f_{g(h,w,i,j,u)}|^p \right)^{1/p}$$ → p-norm subsampling (pooling)

$$g(h,w,i,j,u) = (r\cdot i + h, r\cdot j + w, u)$$

k → pooling size, $k/2$ → half-length, r → stride

$p \to \infty \implies$ max-pooling


conv3

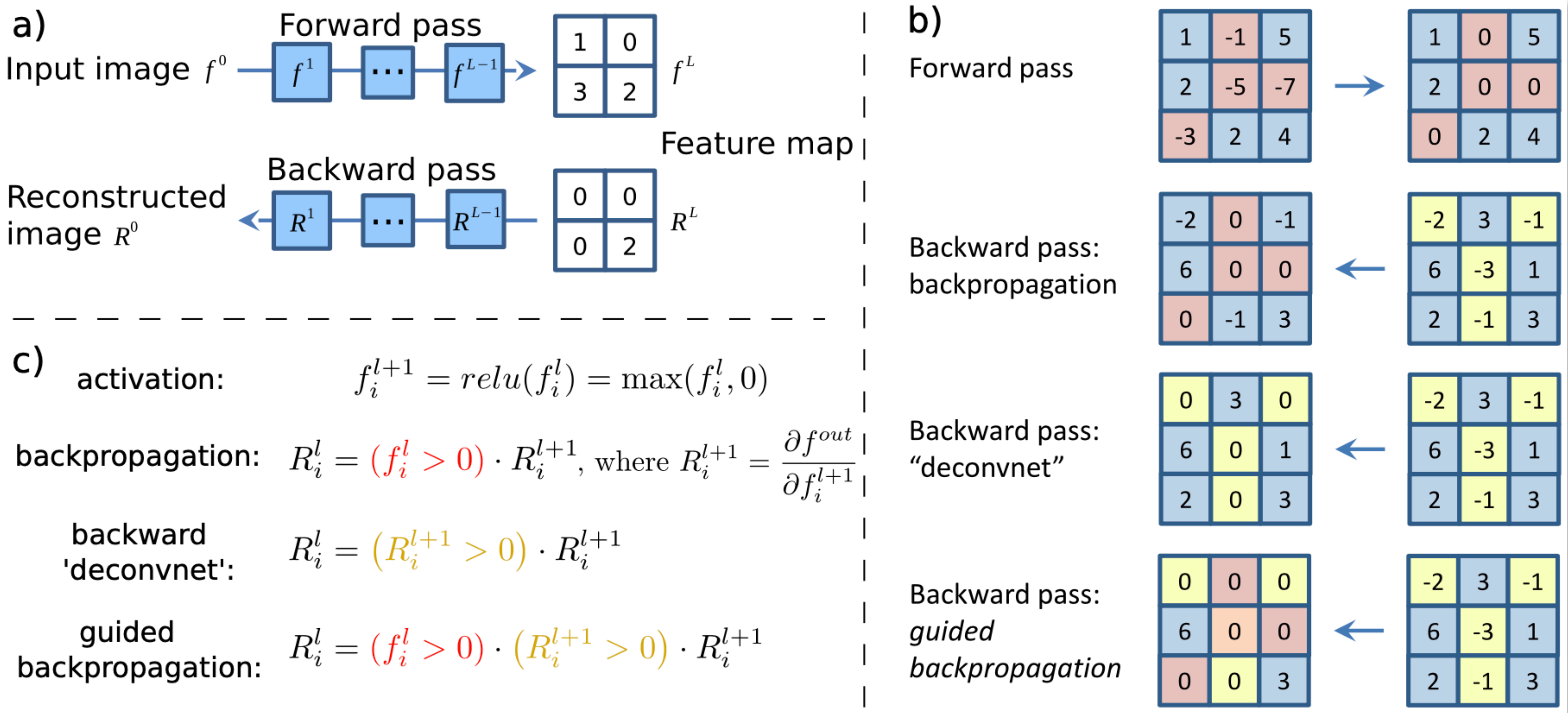
$$c_{i,j,o}(f) = \sigma\left(\sum_{h=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{w=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{u=1}^{N} \theta_{h,w,u,o} \cdot f_{g(h,w,i,j,u)} \right)$$

convolutional weights (or the kernel weights, or filters)

$\sigma(x) = \max(x,0)$, and $o \in [1,M]$

feature-wise (depth-wise) convolution: $\theta_{h,w,u,o} = 1$ if u equals o and zero otherwise

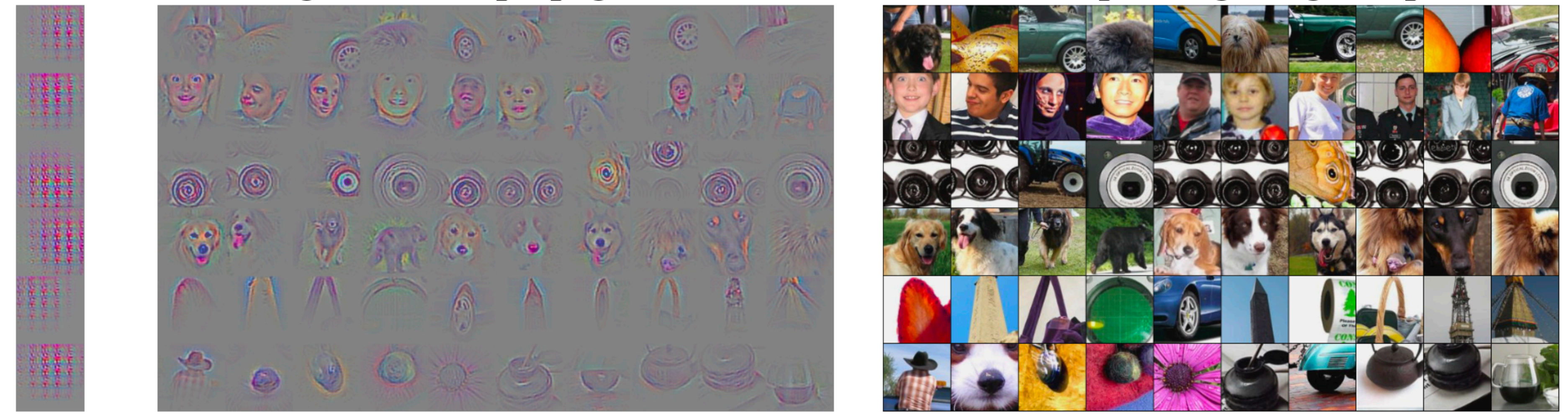
guided backpropagation



Model		
Strided-CNN-C	ConvPool-CNN-C	All-CNN-C
Input 32 × 32 RGB image		
3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU with stride $r = 2$	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU
	3 × 3 max-pooling stride 2	3 × 3 conv. 96 ReLU with stride $r = 2$
3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU with stride $r = 2$	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU
	3 × 3 max-pooling stride 2	3 × 3 conv. 192 ReLU with stride $r = 2$
⋮		

CIFAR-10 classification error		
Model	Error (%)	# parameters
without data augmentation		
Model A	12.47%	≈ 0.9 M
Strided-CNN-A	13.46%	≈ 0.9 M
ConvPool-CNN-A	**10.21%**	≈ 1.28 M
ALL-CNN-A	10.30%	≈ 1.28 M
Model B	10.20%	≈ 1 M
Strided-CNN-B	10.98%	≈ 1 M
ConvPool-CNN-B	9.33%	≈ 1.35 M
ALL-CNN-B	**9.10%**	≈ 1.35 M
Model C	9.74%	≈ 1.3 M
Strided-CNN-C	10.19%	≈ 1.3 M
ConvPool-CNN-C	9.31%	≈ 1.4 M
ALL-CNN-C	**9.08%**	≈ 1.4 M

By using the switches from a forward pass the "deconvnet" (and thereby its reconstruction) is hence conditioned on an image and does not directly visualize learned features. An all convolutional architecture does not include max-pooling, meaning that in theory it can "deconvolve" without switches, i.e. not conditioning on an input image.


deconv　guided backpropagation　corresponding image crops

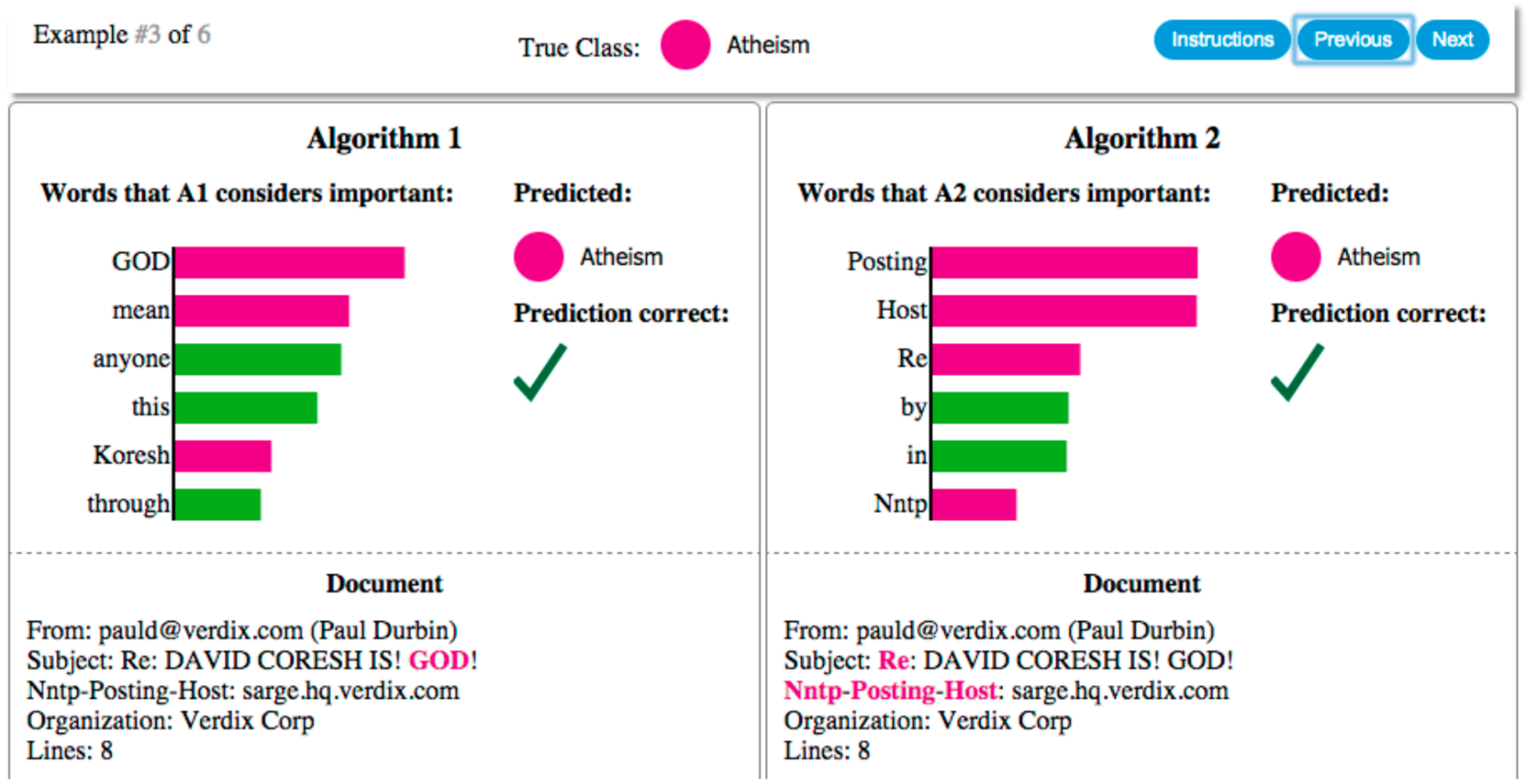
Springenberg, Jost Tobias, et al. "Striving for simplicity: The all convolutional net." *arXiv preprint arXiv:1412.6806* (2014).

“Why Should I Trust You?” Explaining the Predictions of Any Classifier

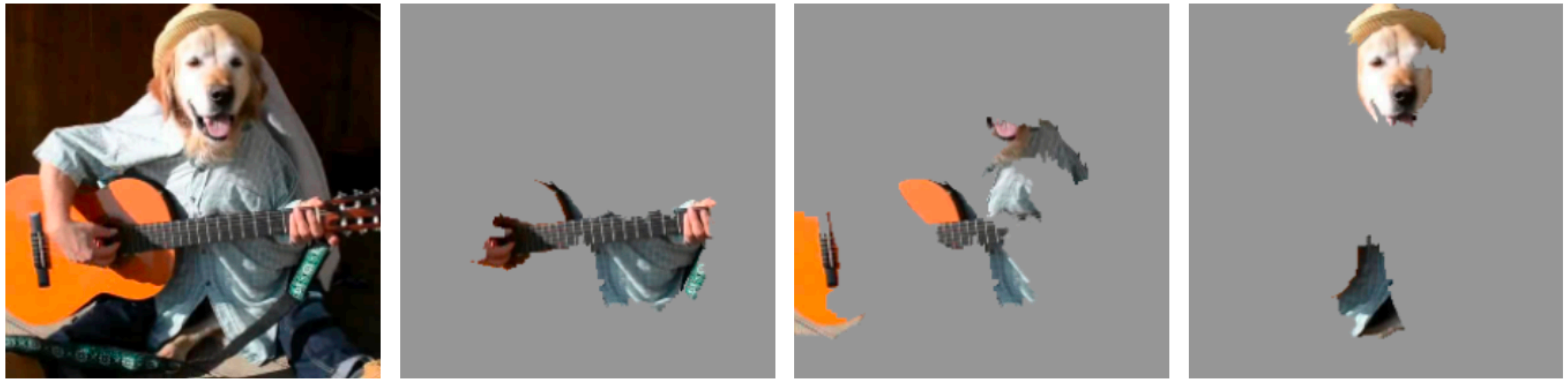
Boulder

YouTube Video

model trained on uni-grams to differentiate “Christianity” from “Atheism”



Google’s Inception Neural Network



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

Local Interpretable Model-agnostic Explanations (LIME)

$x \in \mathbb{R}^d \rightarrow$ original representation of an instance

image: tensor with three color channels per pixel

text: word embeddings

$x' \in \{0,1\}^{d'} \rightarrow$ interpretable representation of an instance

image: “presence” or “absence” of a super-pixel

super-pixel $\rightarrow$ contiguous patch of similar pixels

text: presence or absence of a word

$G \rightarrow$ class of interpretable models

e.g., linear models & decision trees

$g \in G \rightarrow$ explanation

The domain of g is $\{0,1\}^{d'}$

$\Omega(g) \rightarrow$ measure of complexity of model g

e.g., number of non-zero weights in a linear model or depth of a decision tree

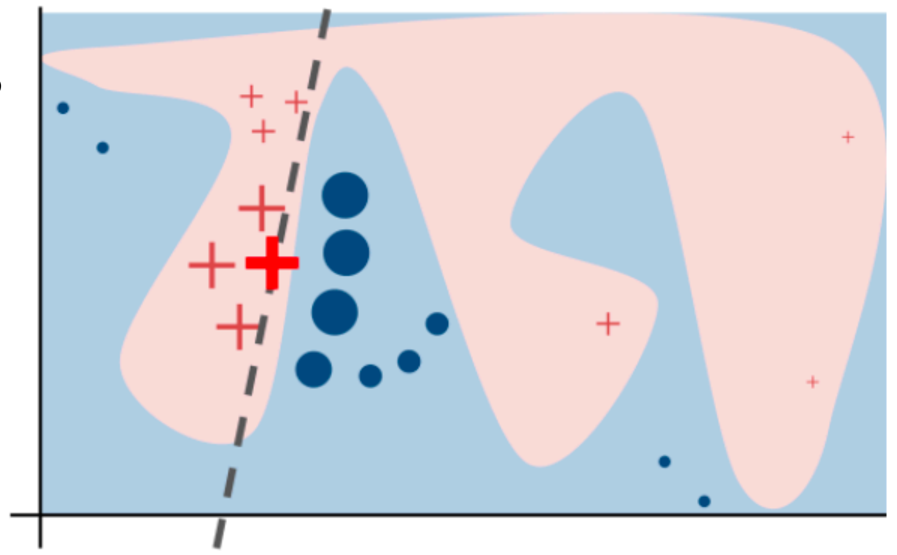
$f : \mathbb{R}^d \rightarrow \mathbb{R}$ $\quad f(x) \rightarrow$ prob. that x belongs to a certain class

model to be explained

$\pi_x(z) \rightarrow$ proximity measure of an instance z to x

$\mathcal{L}(f, g, \pi_x) \rightarrow$ how unfaithful g is in approximating f in the locality defined by π_x

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



Sparse Linear Explanations

$$g(z') = w_g \cdot z' \qquad \Omega(g) = \infty \mathbb{1}[\|w_g\|_0 > K]$$

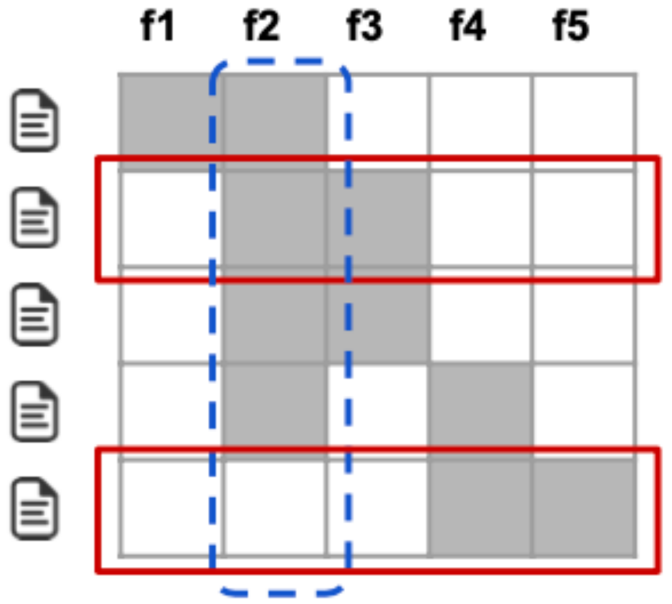
limit on the number of words or super-pixels

$$\pi_x(z) = \exp(-D(x,z)^2/\sigma^2)$$

$D \rightarrow$ cosine distance for text or L_2 distance for images

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

Trusting a model v.s. trusting a prediction



$\mathcal{W} \in \mathbb{R}^{n \times d'}$

$\mathcal{W}_{i,j} = |(w_{g_i})_j|$

Rows (documents) 2 and 5 (in red) would be selected by the pick procedure for the user to inspect.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should I trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.

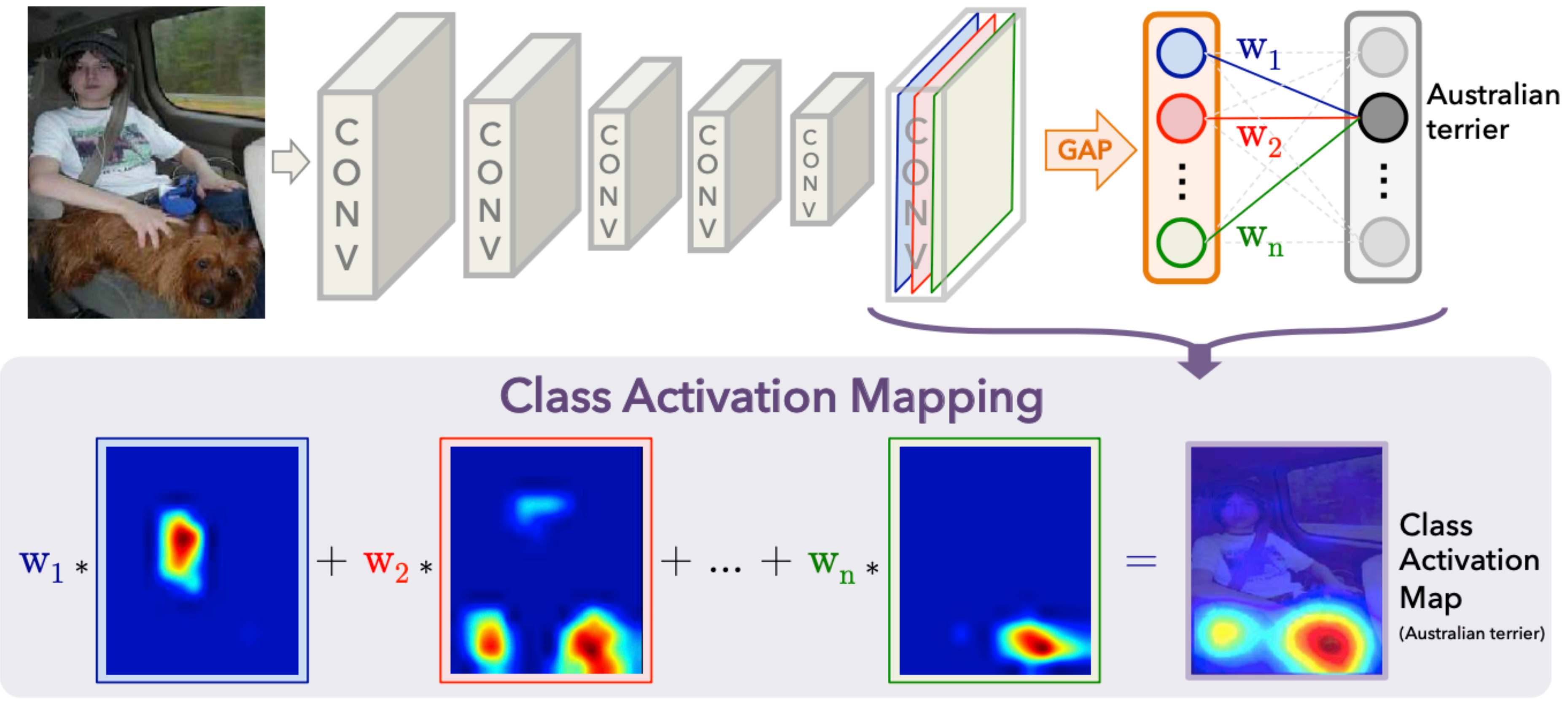
Learning Deep Features for Discriminative Localization

[YouTube Playlist](#)

A class activation map (CAM) for a particular category indicates the discriminative image regions used by the CNN to identify that category.

Brushing teeth

Cutting trees



$f_k(x,y) \rightarrow$ activation of unit k in the last convolutional layer at spatial location (x,y)

$$F_k = \sum_{x,y} f_k(x,y)$$

global average pooling (GAP)

$$S_c = \sum_k w_k^c F_k$$

$S_c \rightarrow$ input to the softmax for a given class c

$w_k^c \rightarrow$ weight corresponding to class c for unit k (importance of F_k for class c)

$$P_c = \frac{\exp(S_c)}{\sum_{c'} \exp(s_{c'})}$$

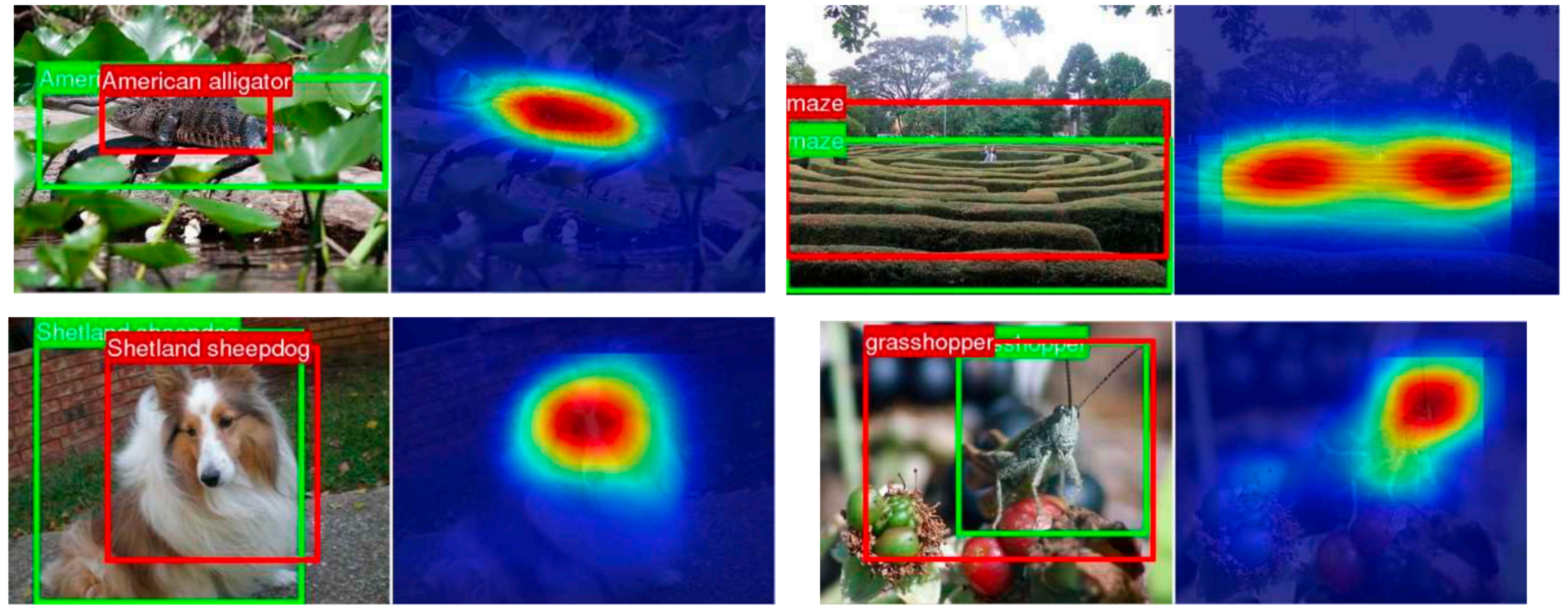
output of the softmax for class c

$$S_c = \sum_k w_k^c \overbrace{\sum_{x,y} f_k(x,y)}^{F_k} = \underbrace{\sum_{x,y} \sum_k w_k^c f_k(x,y)}_{M_c(x,y)}$$

class activation map (CAM)

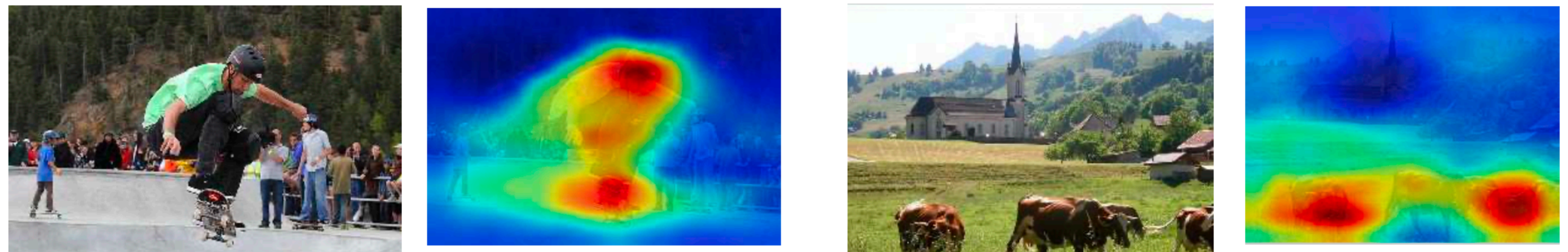
Importance of activation at spatial grid (x,y) leading to the classification of an image to class c.

Weakly-supervised Object Localization



Weakly supervised text detector

Visual question answering



What is the sport?
Prediction: skateboarding

Where are the cows?
Prediction: on the grass

Zhou, Bolei, et al. "Learning deep features for discriminative localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Understanding Deep Learning Requires Rethinking Generalization

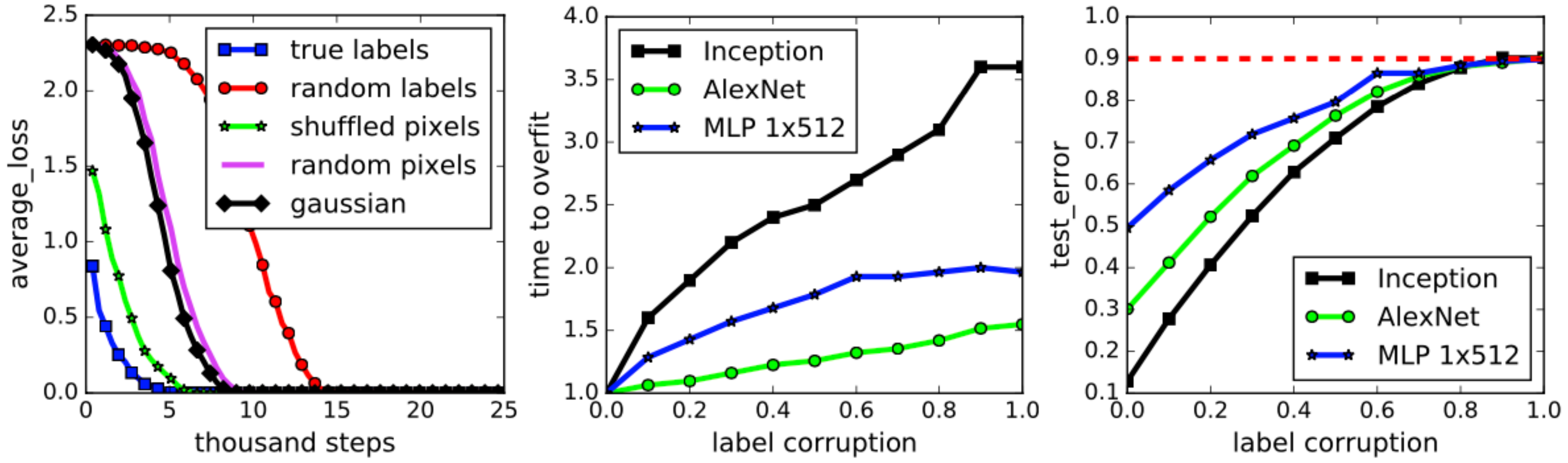
Boulder

YouTube Playlist

generalization error: difference btw "training" & "testing" error

Randomization tests

Deep neural networks easily fit random labels.



(a) learning curves (b) convergence slowdown (c) generalization error growth

The role of explicit regularization

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

The model architecture itself isn't a sufficient regularizer.

Explicit regularization may improve generalization performance, but is neither necessary nor by itself sufficient for controlling generalization error.

Lemma: For any two interleaving sequences of n real numbers $b_1 < x_1 < \ldots < b_n < x_n$, the $n \times n$ matrix $A = [\max(x_i - b_j, 0)]_{ij}$ has full rank. Its smallest eigenvalue is $\min_i (x_i - b_i)$.

proof: A is lower triangular. A is full rank iff all the diagonal entries $\neq 0$.

$x_i > b_i \implies \max(x_i - b_i, 0) > 0 \implies A$ is invertible.

A lower triangular matrix has all its eigenvalues on the main diagonal.

Finite Sample Expressivity

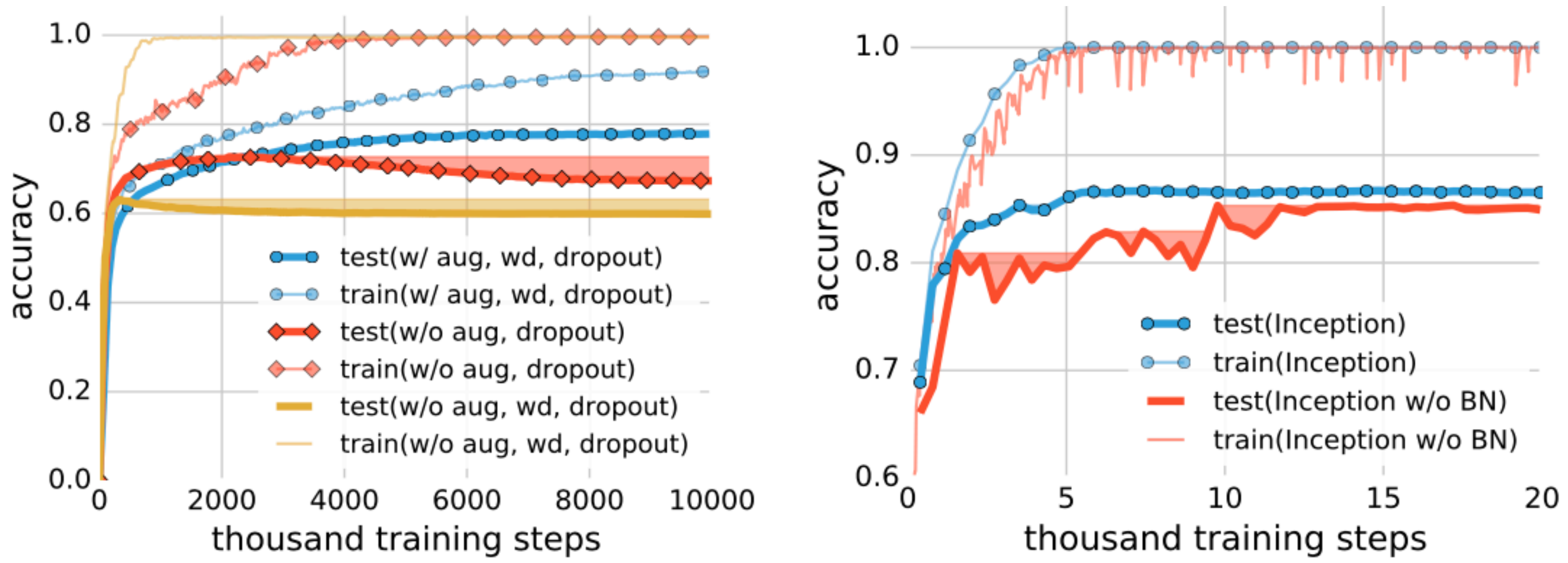
$$c(x) = \sum_{j=1}^{n} w_j \max(\langle a, x\rangle - b_j, 0) \qquad w \in \mathbb{R}^n, b \in \mathbb{R}^n, a \in \mathbb{R}^d \qquad c : \mathbb{R}^d \to \mathbb{R}$$

$S = \{z_1, \ldots, z_n\}$ $\quad y \in \mathbb{R}^n$ $\quad$ Find w, b, a such that $y_i = c(z_i)$ for all $i = 1, \ldots, n$.

Choose a & b such that $x_i := < a, z_i >$ & $b_1 < x_1 < b_2 < x_2 < \ldots < b_b < x_n$.

Theorem: There exists a two-layer neural network with ReLU activations and $2n + d$ weights that can represent any function on a sample of size n in d dimensions

Trade width for depth!



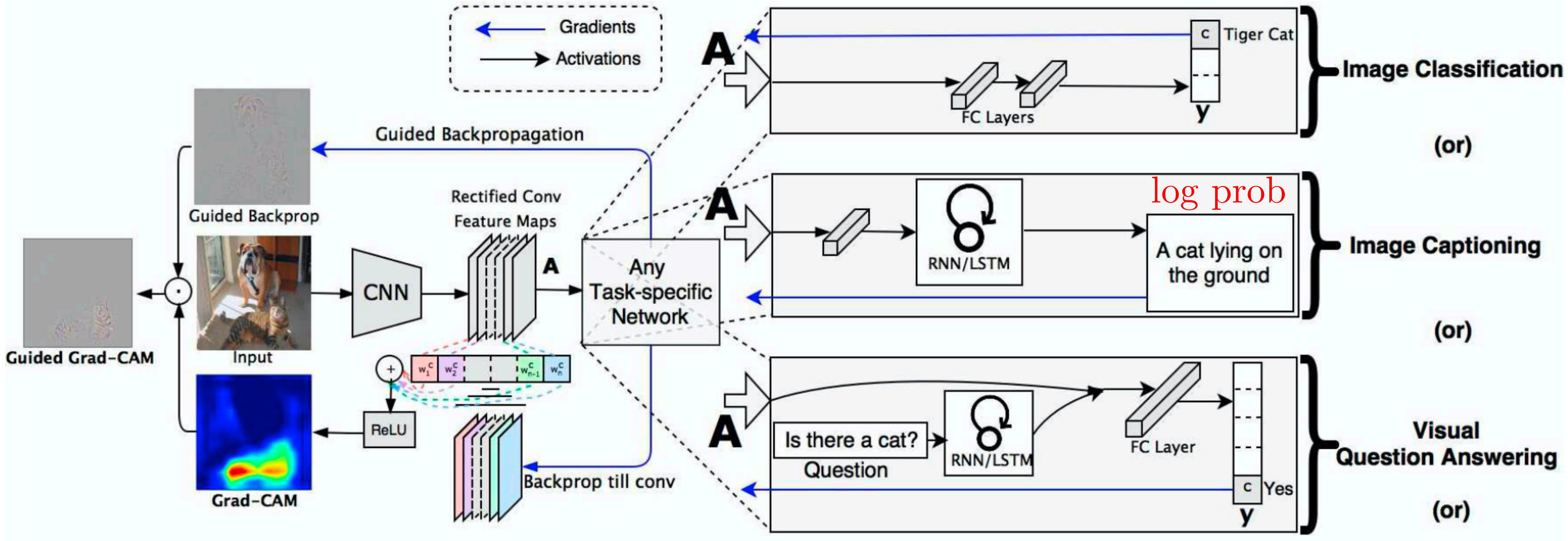
(a) Inception on ImageNet (b) Inception on CIFAR10

Zhang, Chiyuan, et al. "Understanding deep learning requires rethinking generalization." *arXiv preprint arXiv:1611.03530* (2016).

Boulder

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

YouTube Video



Grad-CAM: Gradient-weighted Class Activation Mapping

$L^c_{\text{Grad-CAM}} \in \mathbb{R}^{u \times v}$ class discriminative localization map

$\frac{\partial y^c}{\partial A^k}$ — y^c: score for class c (before softmax); A^k: feature maps of a conv layer

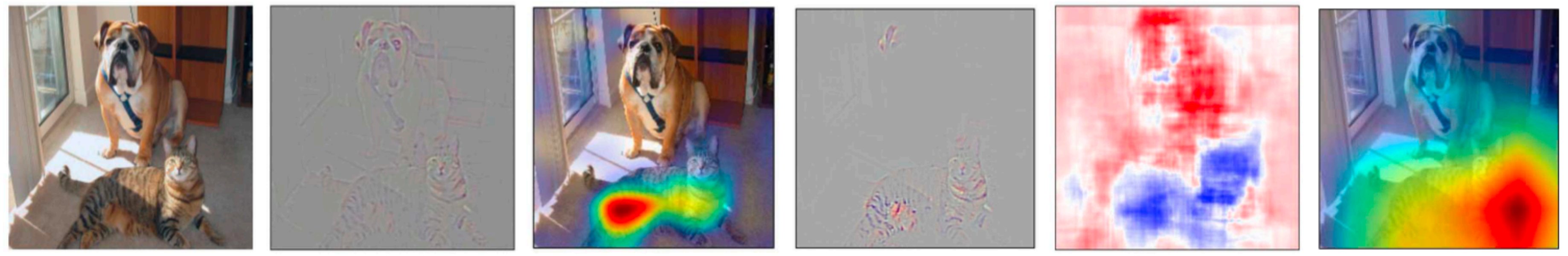
$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

importance weights

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

ReLU: positive influence

a coarse heat map (e.g., 14×14)



(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'

Grad-CAM as a generalization to CAM

$$S^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}$$

$$S^c = \frac{1}{Z}\sum_i \sum_j \underbrace{\sum_k w_k^c A_{ij}^k}_{L^c_{\text{CAM}}}$$

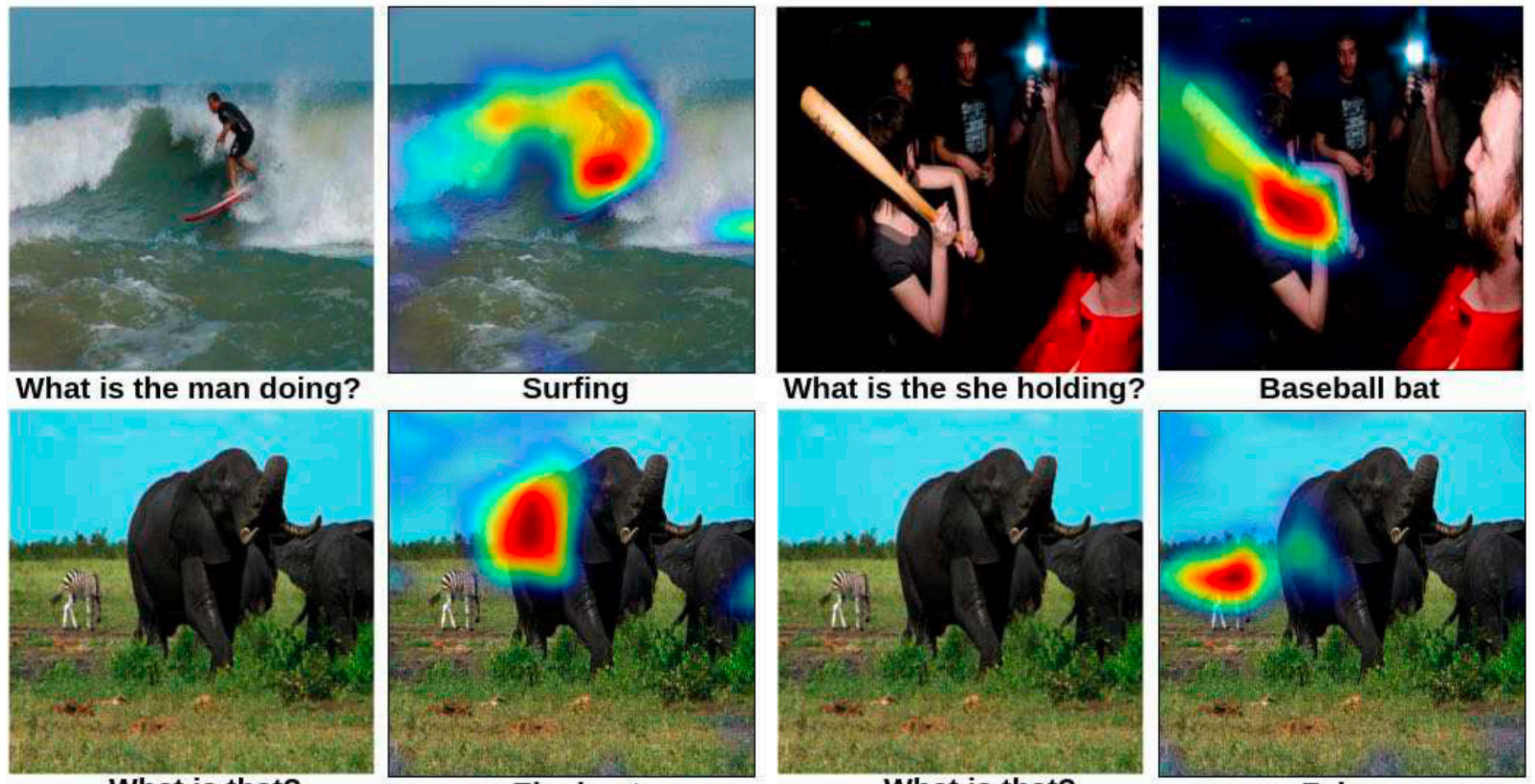
$$\boxed{\alpha_k^c = w_k^c}$$

Image Captioning

A group of people flying kites on a beach

A man is sitting at a table with a pizza

Visual Question Answering



What is the man doing? Surfing

What is the she holding? Baseball bat

What is that? Elephant

What is that? Zebra

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.

A Unified Approach to Interpreting Model Predictions

accuracy-interpretability tradeoff

Additive Feature Attribution Methods

- LIME
- DeepLIFT
- Layer-Wise Relevance Propagation
- Shapley regression values
- Shapley sampling values
- Quantitative Input Influence

$f \rightarrow$ the original prediction model to be explained

$g \rightarrow$ explanation model

local methods: explain a prediction $f(x)$ based on a single input x

$x' \rightarrow$ simplified inputs

$x = h_x(x') \rightarrow$ mapping function

$g(z') \approx f(h_x(z'))$ whenever $z' \approx x'$

$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$ where $z' \in \{0,1\}^M$

$M \rightarrow$ number of simplified input features

SHAP (SHapley Additive exPlanation) Values

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Property 1: Local Accuracy

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

$\phi_0 = f(h_x(0)) \rightarrow$ model output with all simplified inputs toggled off (i.e. missing)

Property 2: Missingness

$$x'_i = 0 \implies \phi_i = 0$$

Property 3: Consistency

$$f^2(h_x(z')) - f^2(h_x(z' \setminus i)) \geq f^1(h_x(z')) - f^1(h_x(z' \setminus i))$$

$$\implies \phi_i(f^2, x) \geq \phi_i(f^1, x)$$

Theorem: g satisfies properties 1, 2, 3 and is unique!

Kernel SHAP (Linear LIME + Shapley values)

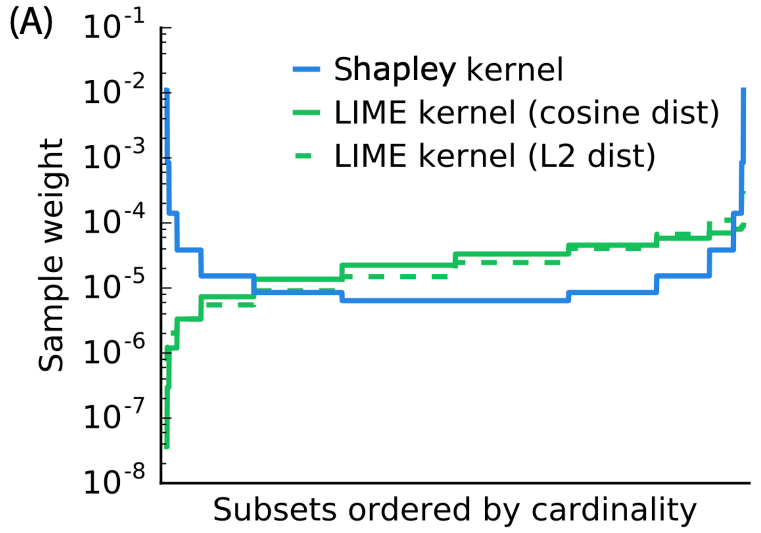
$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$\Omega(g) \rightarrow$ measure of complexity of model g

$\pi_x(z) \rightarrow$ proximity measure of an instance z to x

$\mathcal{L}(f, g, \pi_x) \rightarrow$ how unfaithful g is in approximating f in the locality defined by π_x

$$\boxed{\begin{aligned} \Omega(g) &= 0, \\ \pi_{x'}(z') &= \frac{(M-1)}{(M\ choose\ |z'|)|z'|(M - |z'|)}, \\ L(f, g, \pi_{x'}) &= \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_{x'}(z'), \end{aligned}}$$



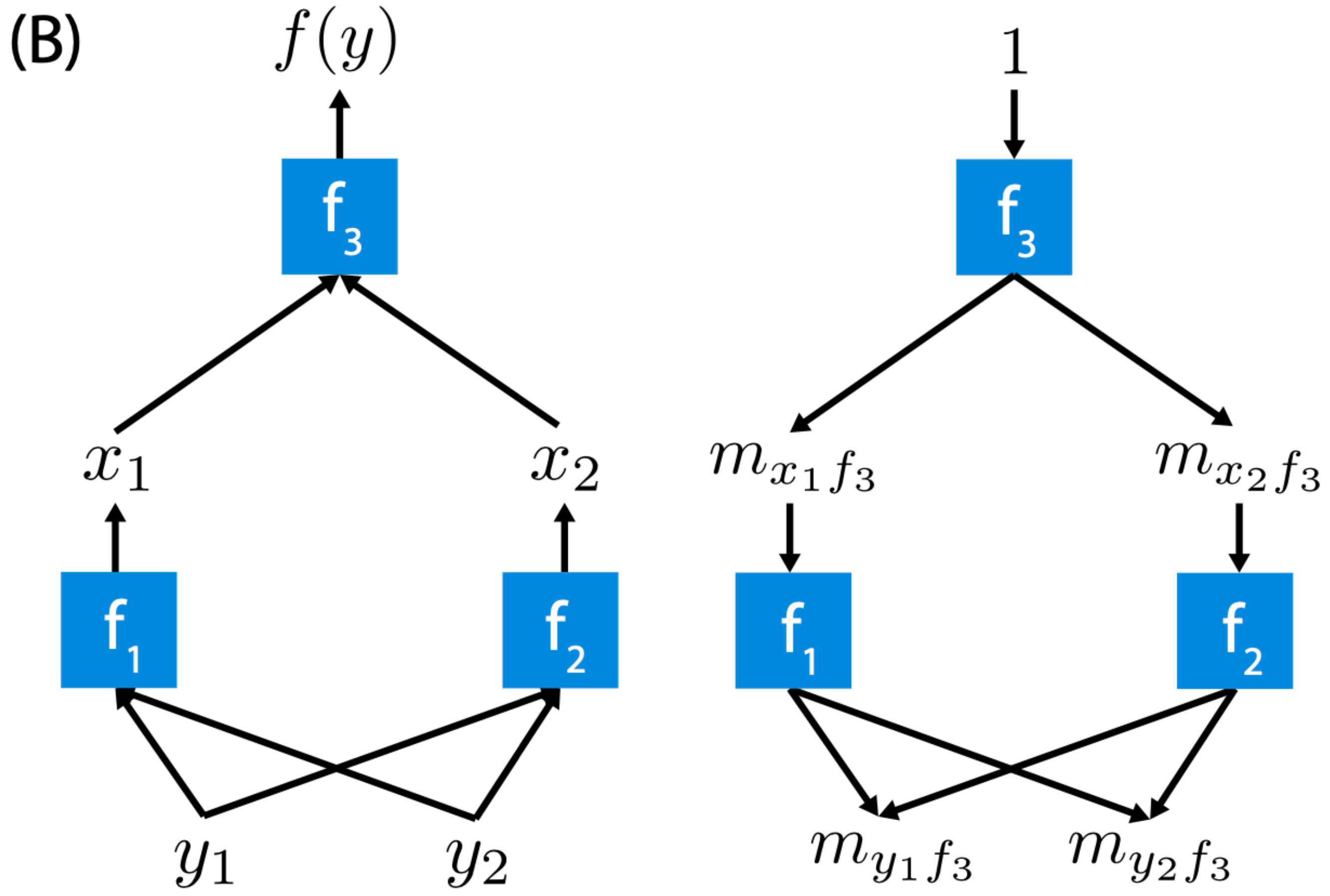
Linear SHAP

$$f(x) = \sum_{j=1}^{M} w_j x_j + b$$

$$\phi_0(f, x) = b$$

$$\phi_i(f, x) = w_j(x_j - E[x_j])$$

Deep SHAP (DeepLIFT + Shapley values)



$$m_{x_j f_3} = \frac{\phi_i(f_3, x)}{x_j - E[x_j]}$$

$$\forall_{j \in \{1,2\}} \quad m_{y_i f_j} = \frac{\phi_i(f_j, y)}{y_i - E[y_i]}$$

$$m_{y_i f_3} = \sum_{j=1}^{2} m_{y_i f_j} m_{x_j f_3}$$

$$\phi_i(f_3, y) \approx m_{y_i f_3}(y_i - E[y_i])$$

The SHAP values for the simple network components can be efficiently solved analytically if they are linear, max pooling, or an activation function with just one input.

Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems*. 2017.

On Calibration of Modern Neural Networks

$X \in \mathcal{X} \rightarrow$ input

$Y \in \mathcal{Y} = \{1, 2, \ldots, K\} \rightarrow$ label

$\pi(X, Y) = \pi(Y|X)\pi(X) \rightarrow$ ground-truth joint distribution

$h \rightarrow$ neural network

$(\hat{Y}, \hat{P}) = h(X) \rightarrow$ class prediction $\hat{Y}$ and associated confidence $\hat{P}$

We would like $\hat{P}$ to be calibrated (i.e., represent a true probability of correctness)

For example, given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.

$$\underbrace{\mathbb{P}}_{\text{over the joint distribution}}\left(\hat{Y} = Y \mid \hat{P} = p\right) = p, \quad \forall p \in [0, 1] \rightarrow \text{perfect calibration}$$

Reliability Diagrams

$B_m \rightarrow$ set of indices of samples whose prediction confidence $\hat{p}$ falls into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}), m = 1, \ldots, M$

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i) \rightarrow \text{consistent and unbiased estimator of } \mathbb{P}(\hat{Y} = Y | \hat{P} \in I_m)$$

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \rightarrow \text{average confidence}$$

Expected Calibration Error (ECE)

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \underbrace{\left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|}_{\text{calibration gap}}$$

$n \rightarrow$ number of samples

Temperature Scaling

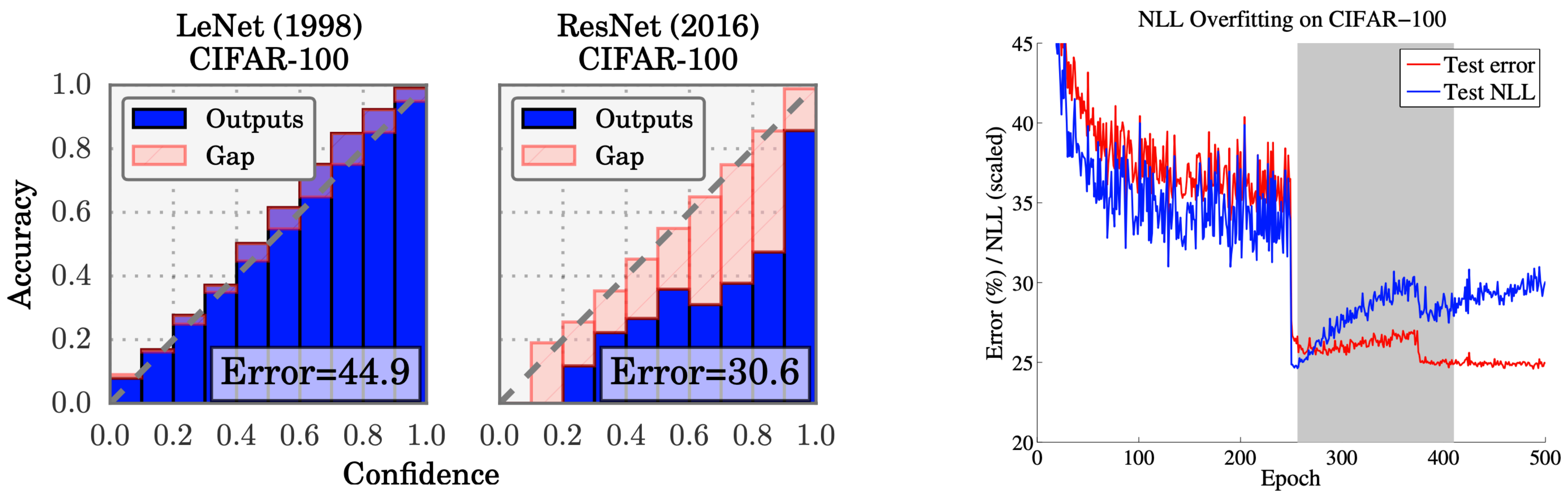
$\mathbf{z}_i \rightarrow$ network logits

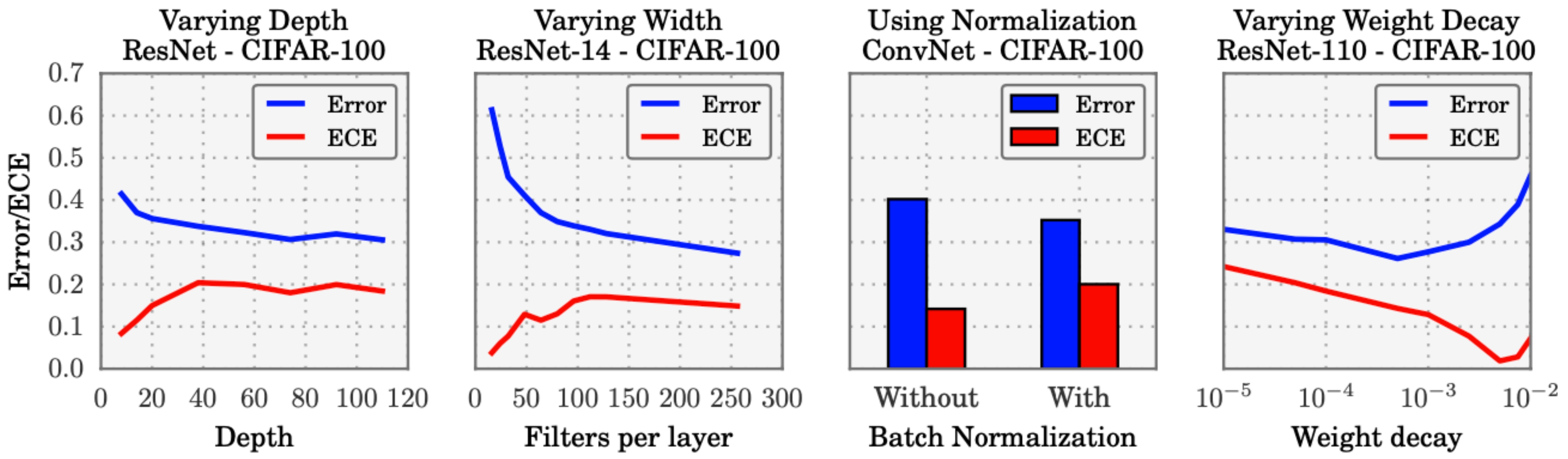
$T > 0 \rightarrow$ temperature scaler

$$\hat{q}_i = \max_k \mathrm{softmax}(\frac{\mathbf{z}_i}{T})^{(k)} \rightarrow \text{calibrated confidence}$$

Optimize T using NLL on the validation data

Modern neural networks are no longer well-calibrated!





Disconnect between NLL (Negative Log Likelihood) and 0/1 loss!

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	9.19%	4.34%	5.22%	4.12%	**1.85%**	3.0%	21.13%
Cars	ResNet 50	4.3%	**1.74%**	4.29%	1.84%	2.35%	2.37%	10.5%
CIFAR-10	ResNet 110	4.6%	0.58%	0.81%	**0.54%**	0.83%	0.88%	1.0%
CIFAR-10	ResNet 110 (SD)	4.12%	0.67%	1.11%	0.9%	**0.6%**	0.64%	0.72%
CIFAR-10	Wide ResNet 32	4.52%	0.72%	1.08%	0.74%	**0.54%**	0.6%	0.72%
CIFAR-10	DenseNet 40	3.28%	0.44%	0.61%	0.81%	**0.33%**	0.41%	0.41%
CIFAR-10	LeNet 5	3.02%	1.56%	1.85%	1.59%	**0.93%**	1.15%	1.16%
CIFAR-100	ResNet 110	16.53%	2.66%	4.99%	5.46%	**1.26%**	1.32%	25.49%
CIFAR-100	ResNet 110 (SD)	12.67%	2.46%	4.16%	3.58%	0.96%	**0.9%**	20.09%
CIFAR-100	Wide ResNet 32	15.0%	3.01%	5.85%	5.77%	**2.32%**	2.57%	24.44%
CIFAR-100	DenseNet 40	10.37%	2.68%	4.51%	3.59%	1.18%	**1.09%**	21.87%
CIFAR-100	LeNet 5	4.85%	6.48%	2.35%	3.77%	**2.02%**	2.09%	13.24%
ImageNet	DenseNet 161	6.28%	4.52%	5.18%	3.51%	**1.99%**	2.24%	-
ImageNet	ResNet 152	5.48%	4.36%	4.77%	3.56%	**1.86%**	2.23%	-
SVHN	ResNet 152 (SD)	0.44%	**0.14%**	0.28%	0.22%	0.17%	0.27%	0.17%
20 News	DAN 3	8.02%	**3.6%**	5.52%	4.98%	4.11%	4.61%	9.1%
Reuters	DAN 3	0.85%	1.75%	1.15%	0.97%	0.91%	**0.66%**	1.58%
SST Binary	TreeLSTM	6.63%	1.93%	**1.65%**	2.27%	1.84%	1.84%	1.84%
SST Fine Grained	TreeLSTM	6.71%	2.09%	**1.65%**	2.61%	2.56%	2.98%	2.39%

Guo, Chuan, et al. "On calibration of modern neural networks." *International Conference on Machine Learning*. PMLR, 2017.

Boulder

Do Vision Transformers See Like Convolutional Neural Networks?

Representation Similarity and CKA (Centered Kernel Alignment)

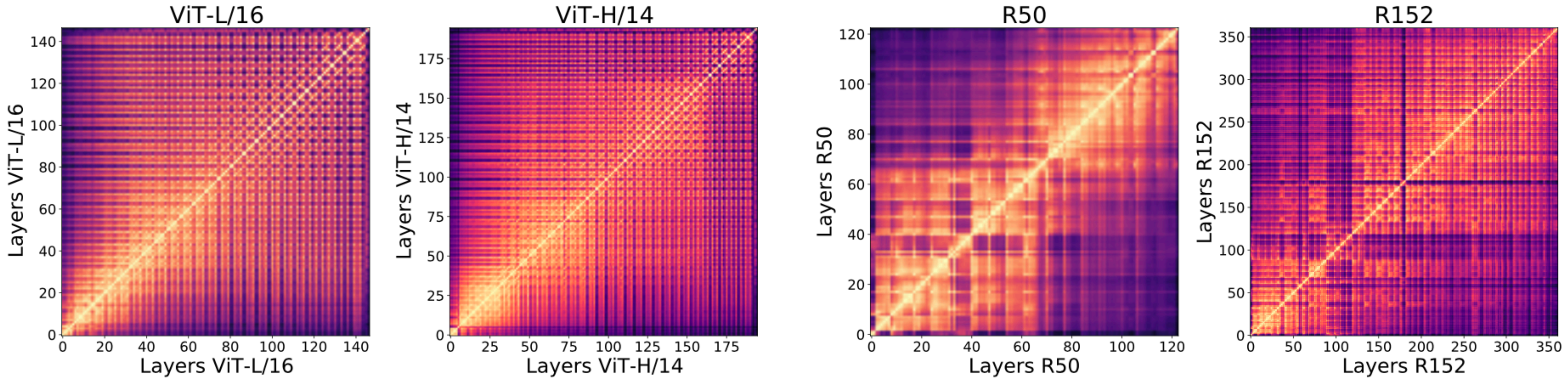
$X \in \mathbb{R}^{m \times p_1}, Y \in \mathbb{R}^{m \times p_2} \rightarrow$ internal representations (i.e., activation matrices) of two layers within one network or across two networks, evaluated at the same examples

$m \rightarrow$ number of examples

$p_1, p_2 \rightarrow$ number of neurons

$K = XX^T, L = YY^T \rightarrow$ Gram matrices (measures the similarity of a pair of datapoints according to layer representations)

$$\text{CKA}(K, L) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K)\text{HSIC}(L, L)}}$$



HSIC $\rightarrow$ Hilbert-Schmidt Independence Criterion

$H = I_n - \frac{1}{n}\mathbb{1}\mathbb{1}^T \rightarrow$ centring matrix

$K' = HKH, L' = HLH \rightarrow$ centered Gram matrices

$\text{HSIC}(K, L) = \text{vec}(K') \cdot \text{vec}(L')/(m-1)^2 \rightarrow$ similarity between the centered Gram matrices

CKA is invariant to orthogonal transformation of representations (including permutation of neurons), and the normalization term ensures invariance to isotropic scaling.

Findings

– ViTs have more uniform representations, with greater similarity between lower and higher layers

– ViT incorporates more global information than ResNet at lower layers, leading to quantitatively different features
  ResNet effective receptive fields are highly local and grow gradually; ViT effective receptive fields shift from local to global
  effective receptive field of different layers: absolute value of the gradient of the center location of the feature map with respect to the input

– incorporating local information at lower layers remains vital, with large-scale pre-training data helping early attention layers learn to do this

– skip connections in ViT are even more influential than in ResNets, having strong effects on performance and representation similarity

– motivated by potential future uses in object detection, the paper examines how well input spatial information is preserved

– the paper studies the effects of dataset scale on transfer learning, with a linear probes study revealing its importance for high quality intermediate representations

Raghu, Maithra, et al. "Do Vision Transformers See Like Convolutional Neural Networks?." *arXiv preprint arXiv:2108.08810* (2021).

Questions?