Visualization of DNN

2019.12.16 熊凯

OverView

Mainly introduce two kinds of methods (注：后面按照方法间的相关性展开介绍)

- DeConv-Based
  1. DeConvNet2014
  2. Invert2016
- Gradient-Based
  1. DeepInside2014 (EasyFool2015)
  2. (Guided-) GradCAM2017 & GradCAM++2018
  3. GuidedBackprop2015
  4. Invert2015
- Others

  CAM2016

  RISE2018 (Perturb2017)
- Applications

Related Topics:

Visualization

Saliency

weakly supervised localization

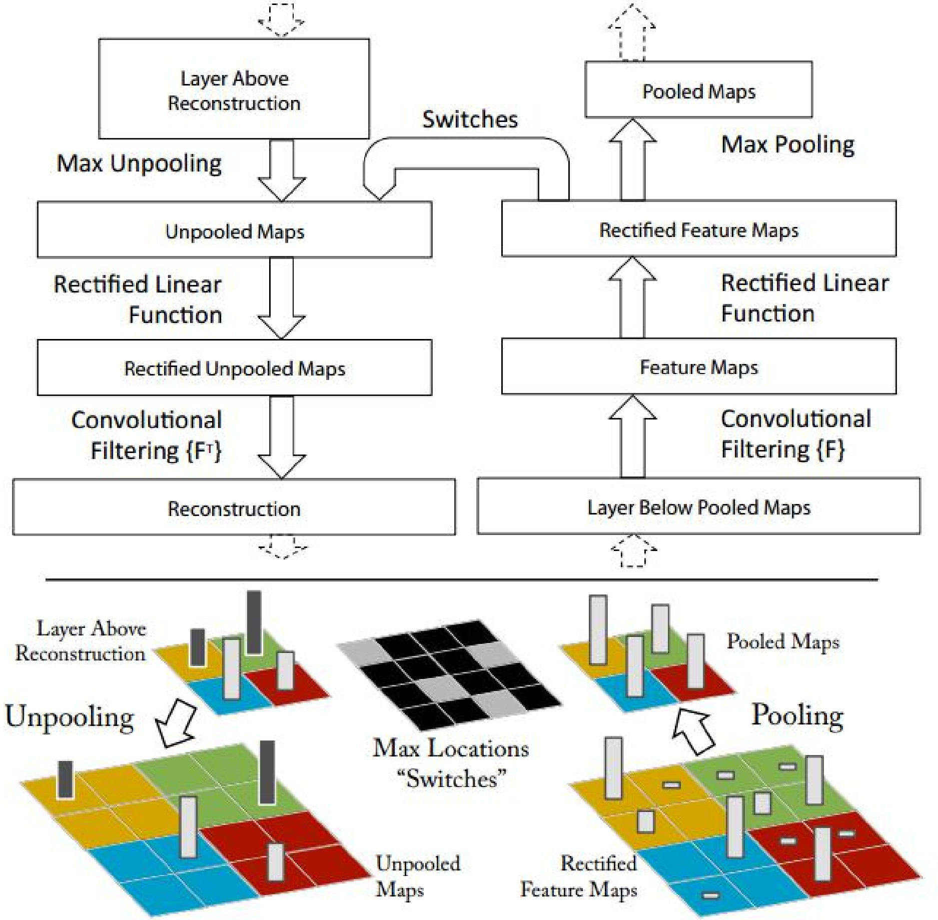
weakly supervised segmentation

Explanation/Interpretation/attribution

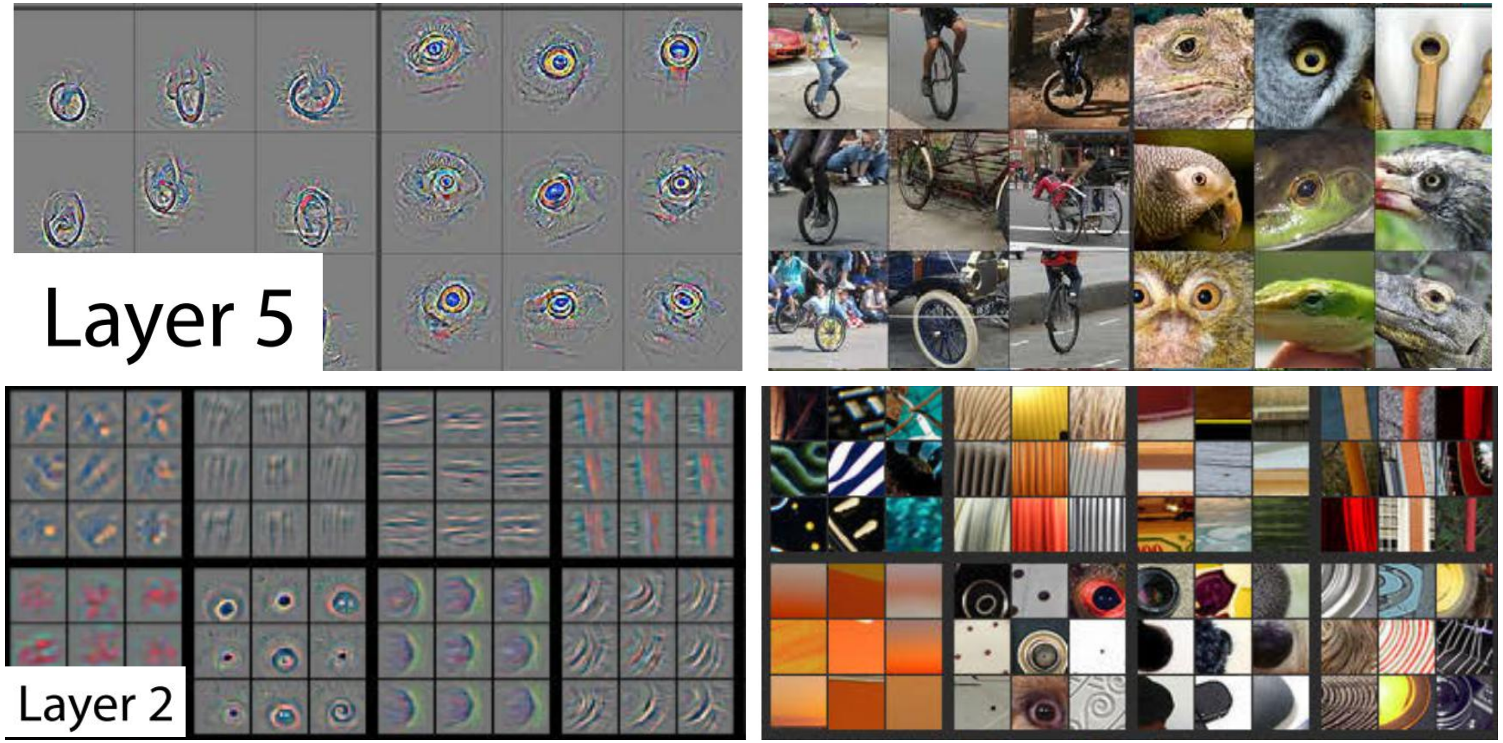
Methods

DeConvNet2014

ECCV2014, Visualizing and Understanding Convolutional Networks



Top: A deconvnet layer(left) attached to a convnet layer(right). **Bottom:** An illustration of the unpooling operation, using switches which record the location of the local max.



展示了两个layer的部分特征图及对应的图像块，**3*3小方格中对应了特征图的同一通道**在验证集上的top-9激活，分别进行deconv后的可视化结果

- **基于DeConv、UnPolling，**将特征图的最大激活值反传(即将其他所有元素置0)，**重构输入，**作为特定激活单元学到的特征。属Image-specific!
- 须专门定义deconv模型

DeepInside2014

对抗样本/DeepDream: 类别生成图的输入由随机噪声变为真实图片

ICLR2014, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps -- 可看做DeConvNet2014的泛化

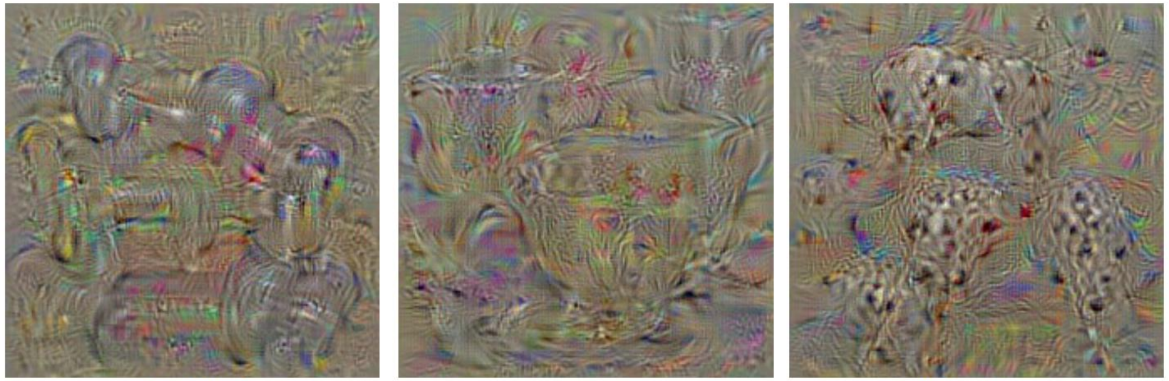

dumbbell
cup
dalmatian

Figure 1: **the class appearance models**

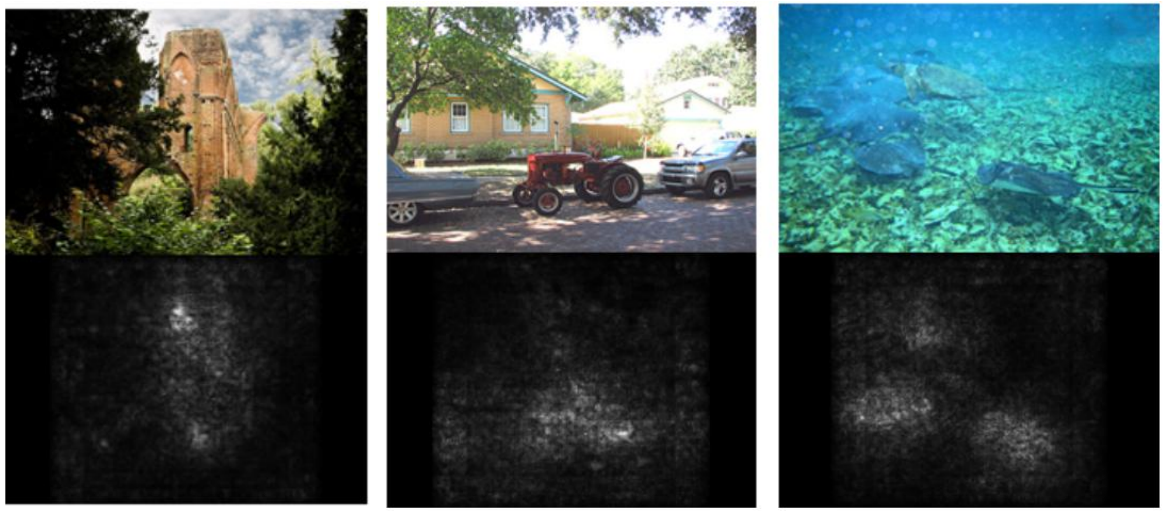


Figure 2: **Image-specific class saliency maps**

可结合GraphCut实现弱监督目标定位

Class Model Visualisation

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2, \quad (1)$$

对输入求导并更新，看使特定类得分最大的输入长啥样

若不加正则？

Image-Specific Class Saliency Visualisation

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}. \quad (4)$$

对输入求导，看输入中哪些位置对特定类得分贡献最大，作为显著图

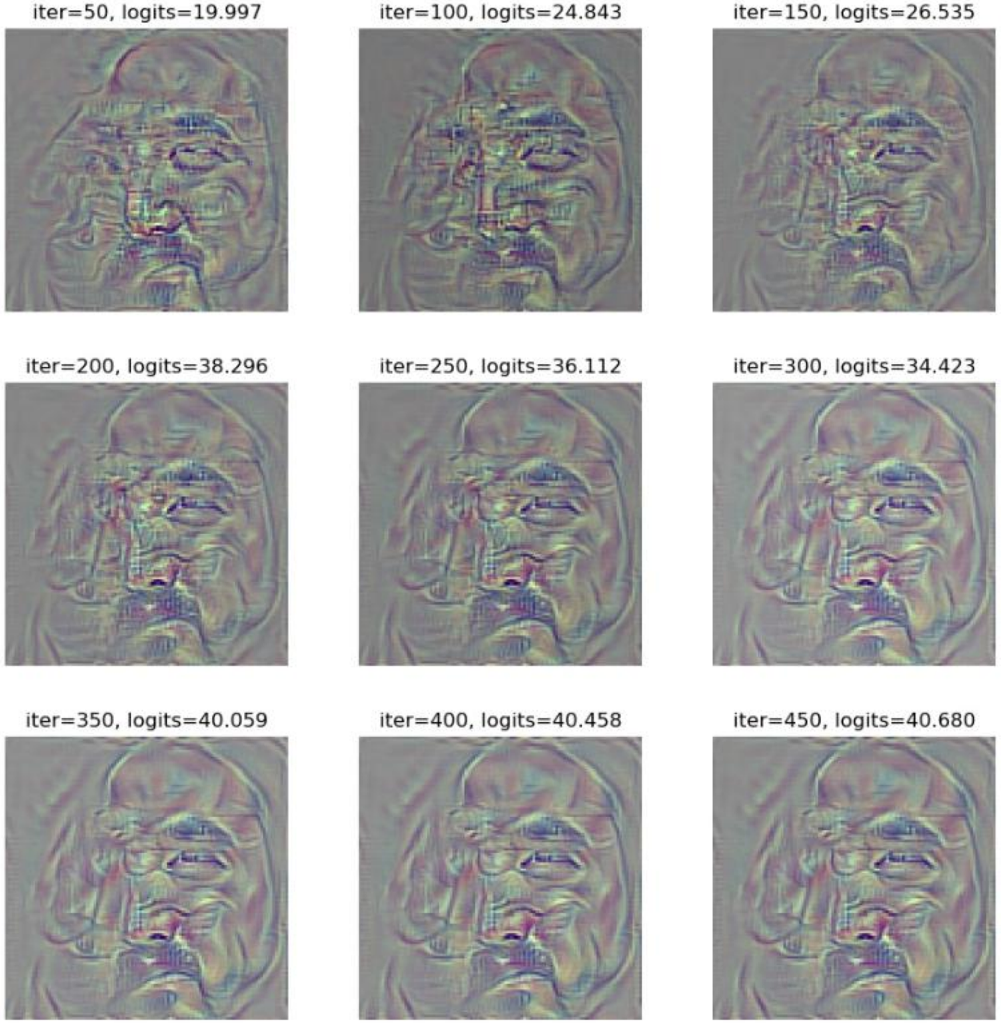
CVPR2015(DNN are easily fooled)指出, 不加正则可生成unrecognizable但置信度很高的图像，加正则后图更recognizable但类别置信度降低。

思考：底层特征允许重构单个样本；高层特征/类向量要关注不变性，所以特定的某个样本，并不能使其最大激活，故Class Model Vis学到的也应该是某种“共性”如平均脸。

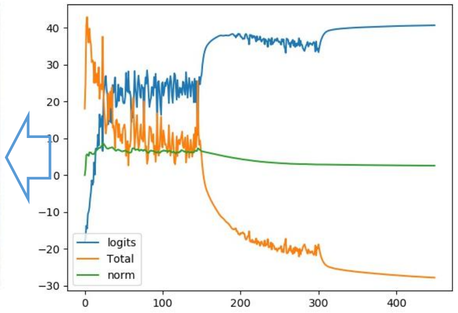
- **可看做DeConvNet2014的泛化，只有ReLU层有差异。**（基于梯度的优势在于能可视化任一层而非仅conv层；后面Guided-Backprop2015中会提到ReLU差异)

实验 20191119-1121

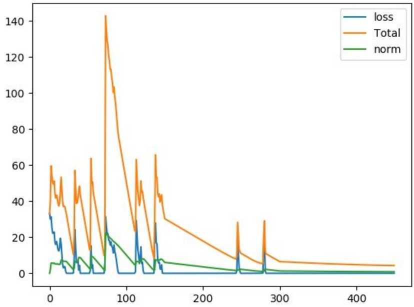
class model visualization

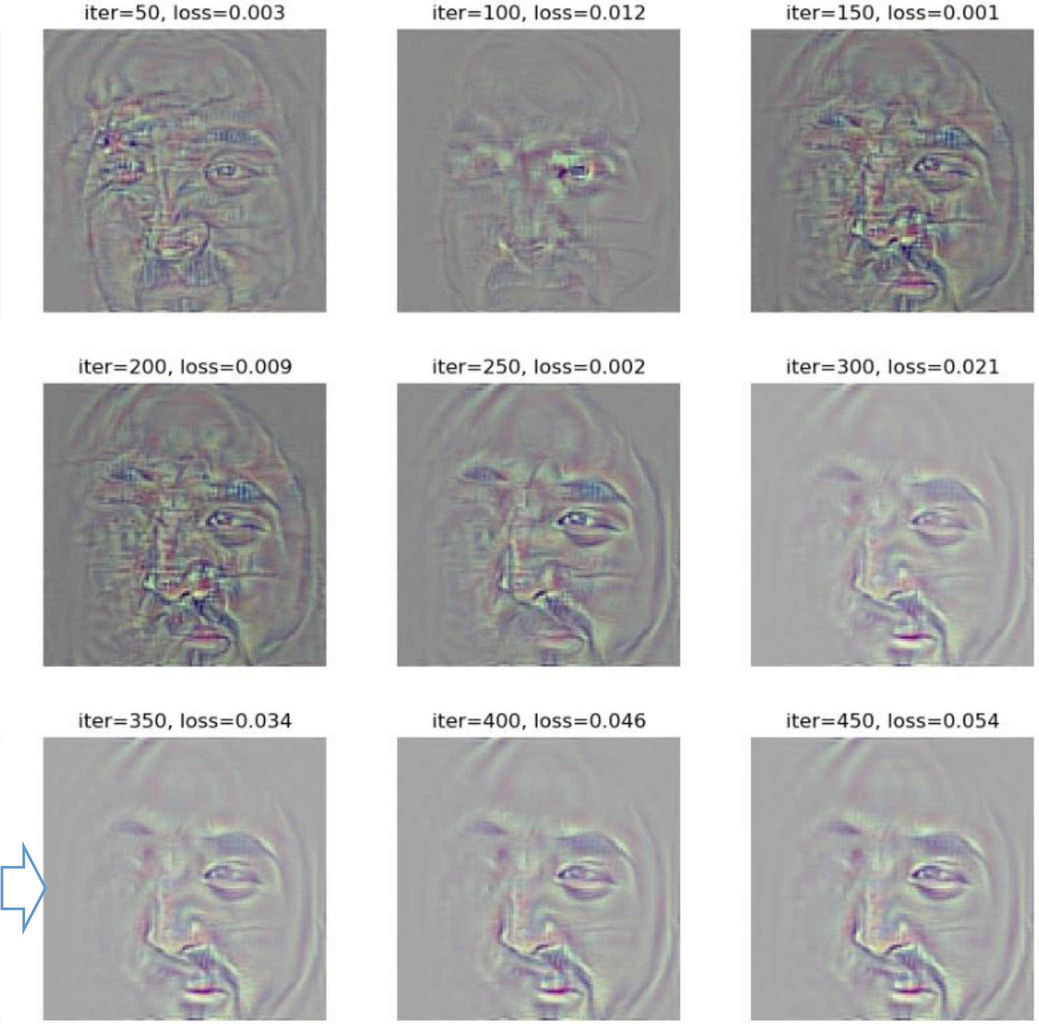


logit-based(left)



label-34340





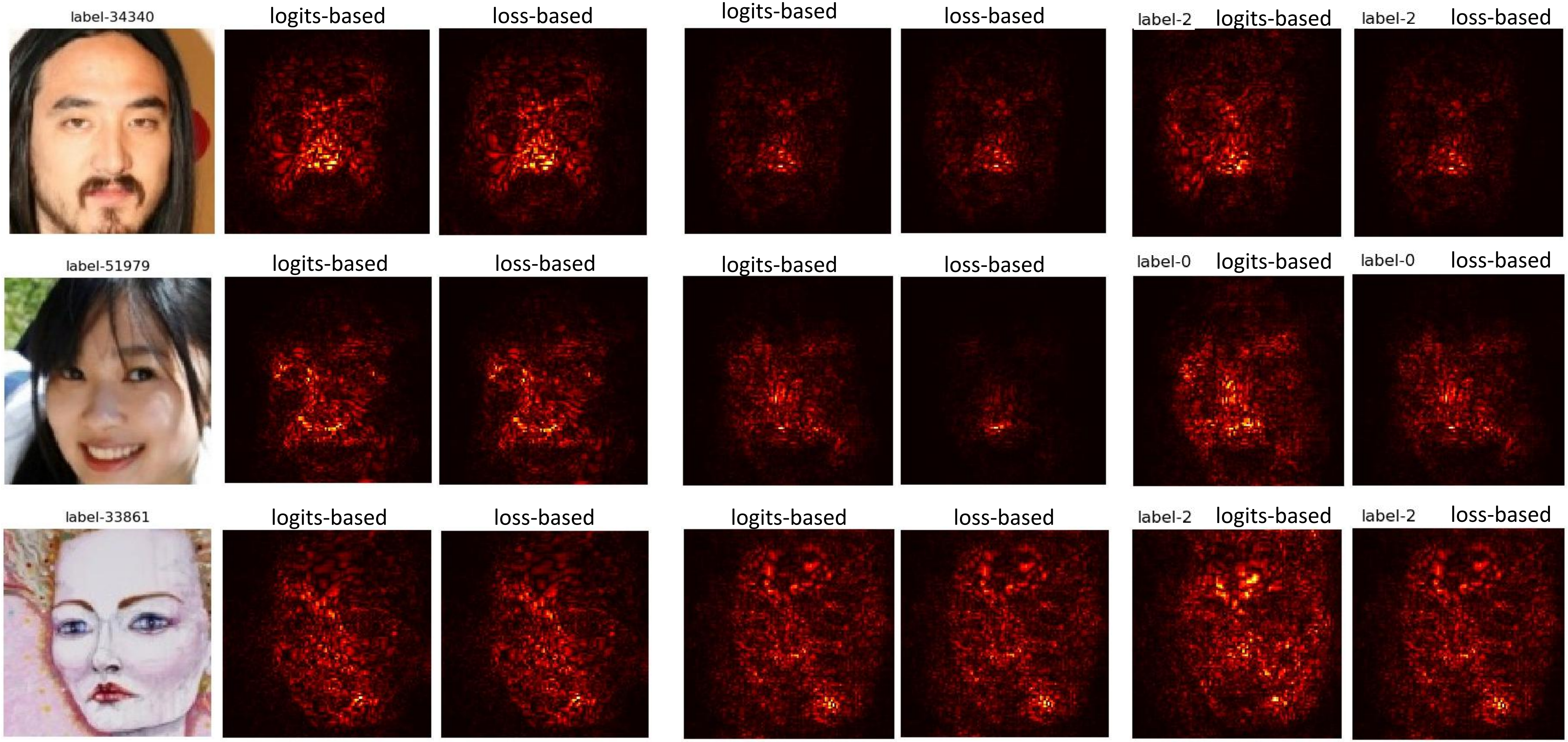
loss-based(right)

params:λ=5, lr=0.1,lr_stage=[150,300]; model: IR_50_11.1

实验 20191119-1121

class saliency visualization

- image specific $S_c(I) \approx w^T I + b, \quad w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}$

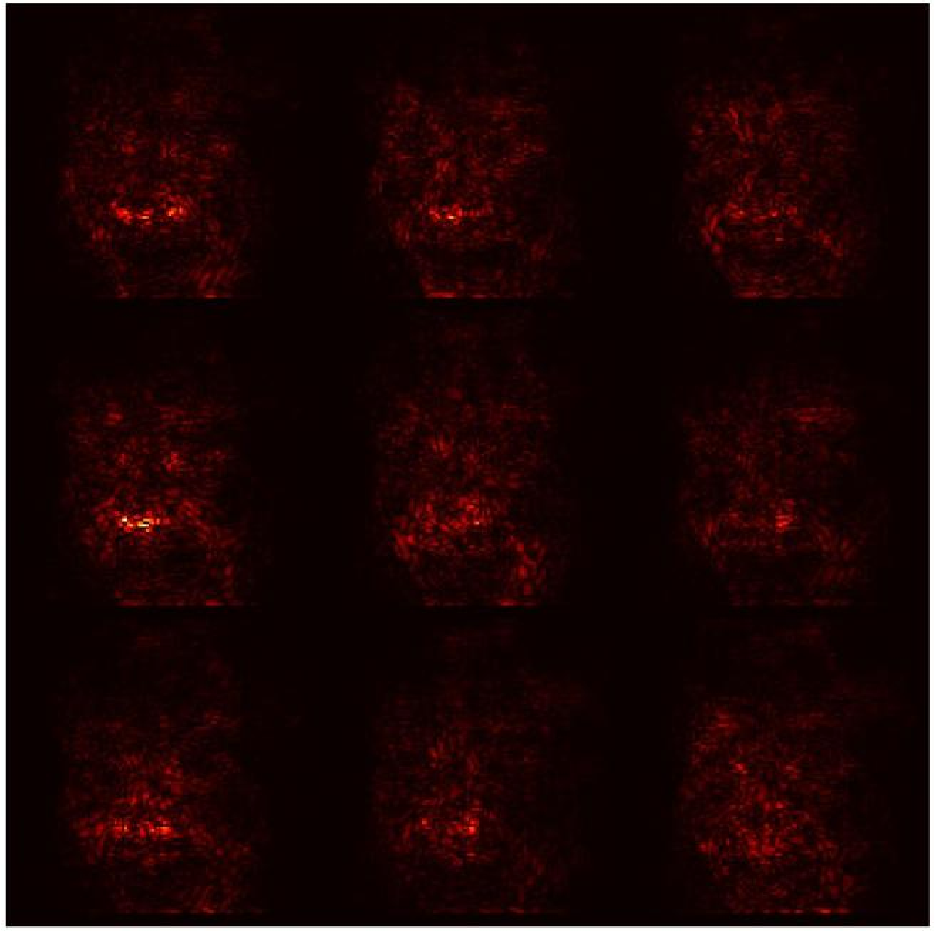


IR_50_11.1 **MobNet_10.22** **MobNet_10.22(wrong_label)**

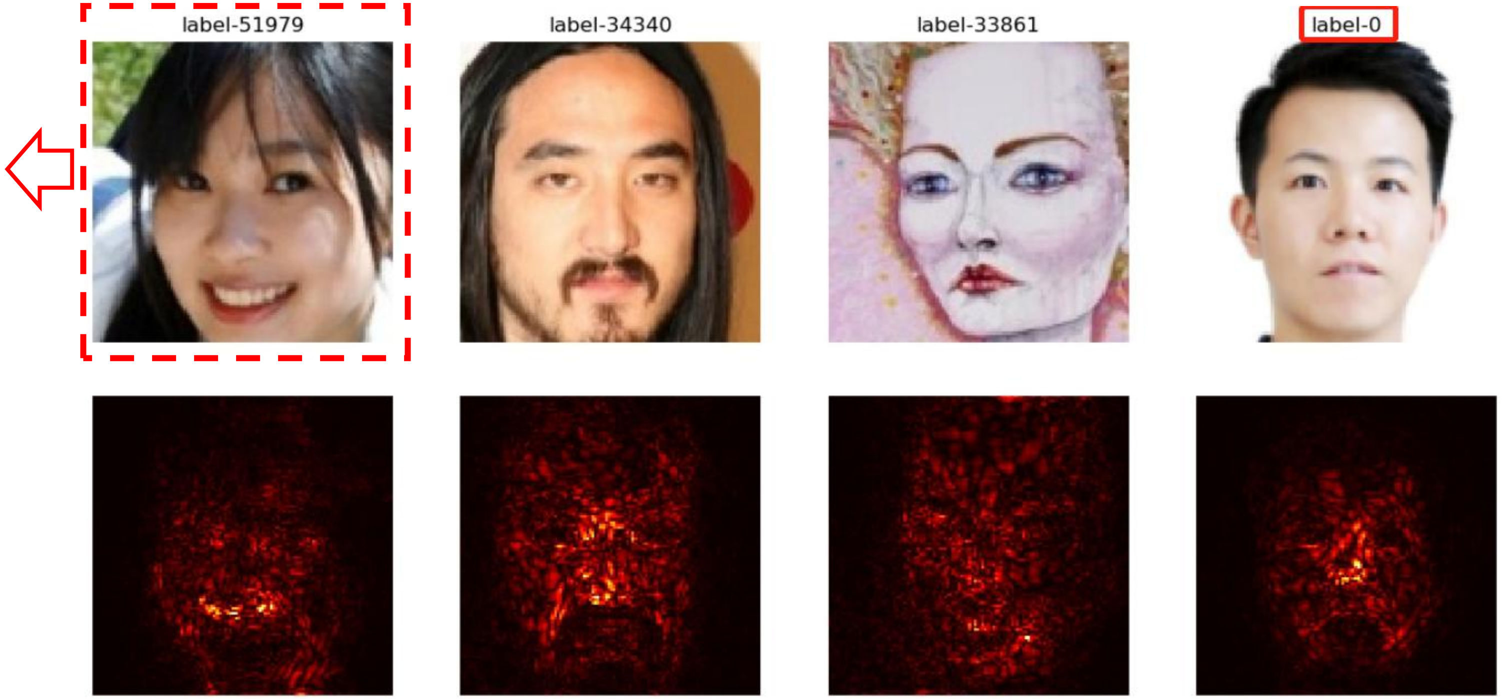
实验 20191119-1121

扩展：feature saliency visualization

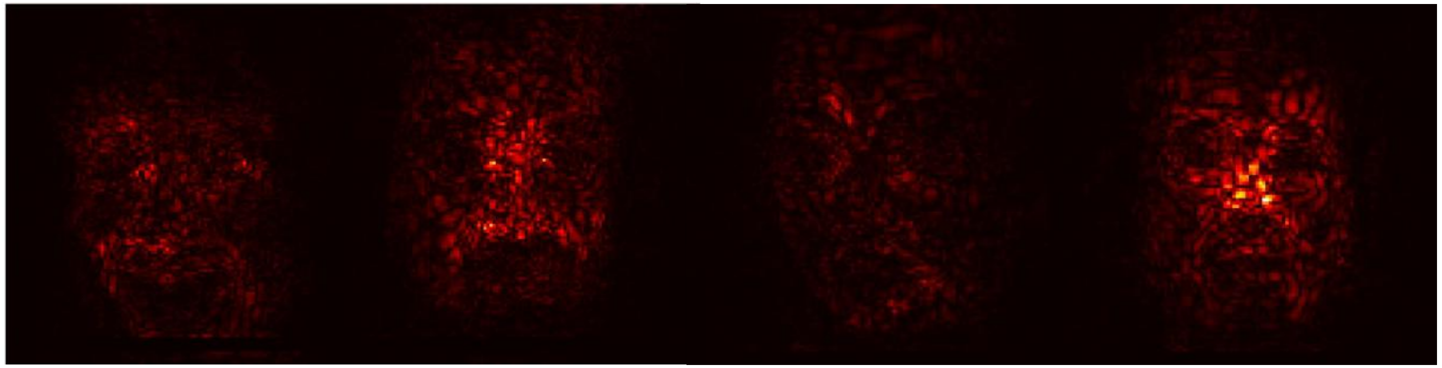
- generate image



512维特征中**第1~9维特征**对应的saliency map



各图输出的512维特征中**第1维特征**对应的saliency map



各图输出的512维特征对应的merged saliency map(基于梯度累加)

支持可视化：

1.单图多通道；2.多图单通道；3.多图merged通道

IR_50_11.1

CAM2016

CVPR2016, MIT, Learning Deep Features for Discriminative Localization

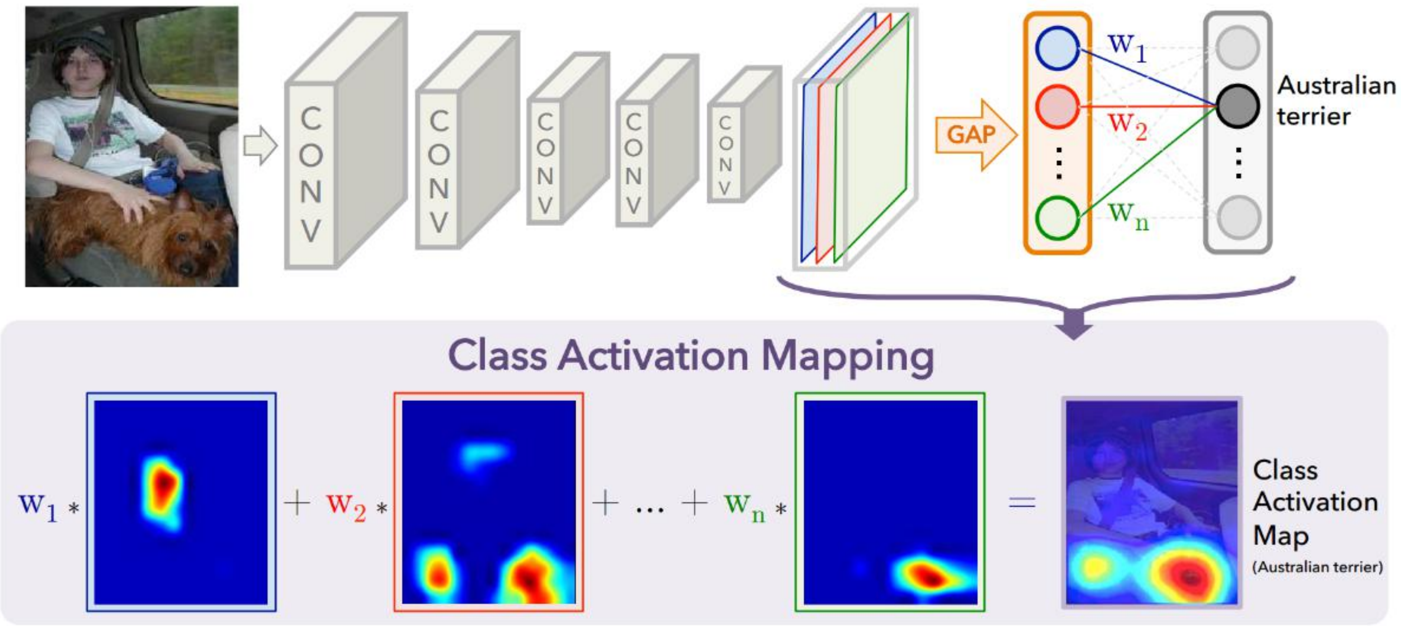


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the cla activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

- GMP vs GAP

1. When doing GMP the value can be maximized by finding all discriminative parts of an object as all low activations reduce the output of the map.
2. **GMP achieves similar classification performance as GAP, GAP outperforms GMP for localization**.

注：激活图直接上采样到原图大小

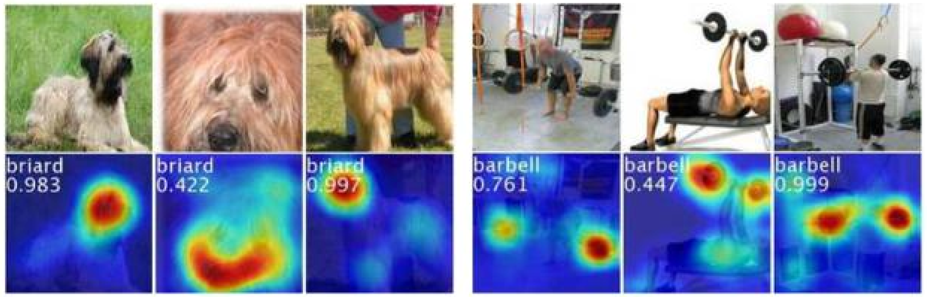


Figure 3. The CAMs of two classes from ILSVRC [21]. The maps highlight the discriminative image regions used for image classification, the head of the animal for *briard* and the plates in *barbell*.

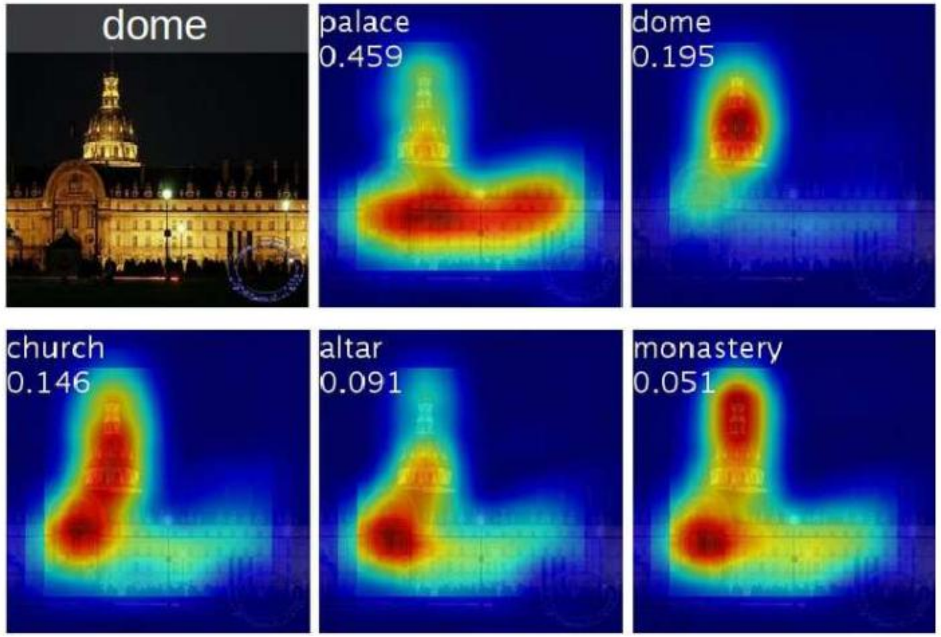


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome.

GradCAM2017

https://github.com/ramprs/grad-cam/,
http://gradcam.cloudcv.org

as generalization of CAM

arixv2017, Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

$$L^c_{\text{CAM}} = \sum_k \underbrace{w^c_k}_{\text{class feature weights}} A^k$$

CAM VS. Grad-CAM

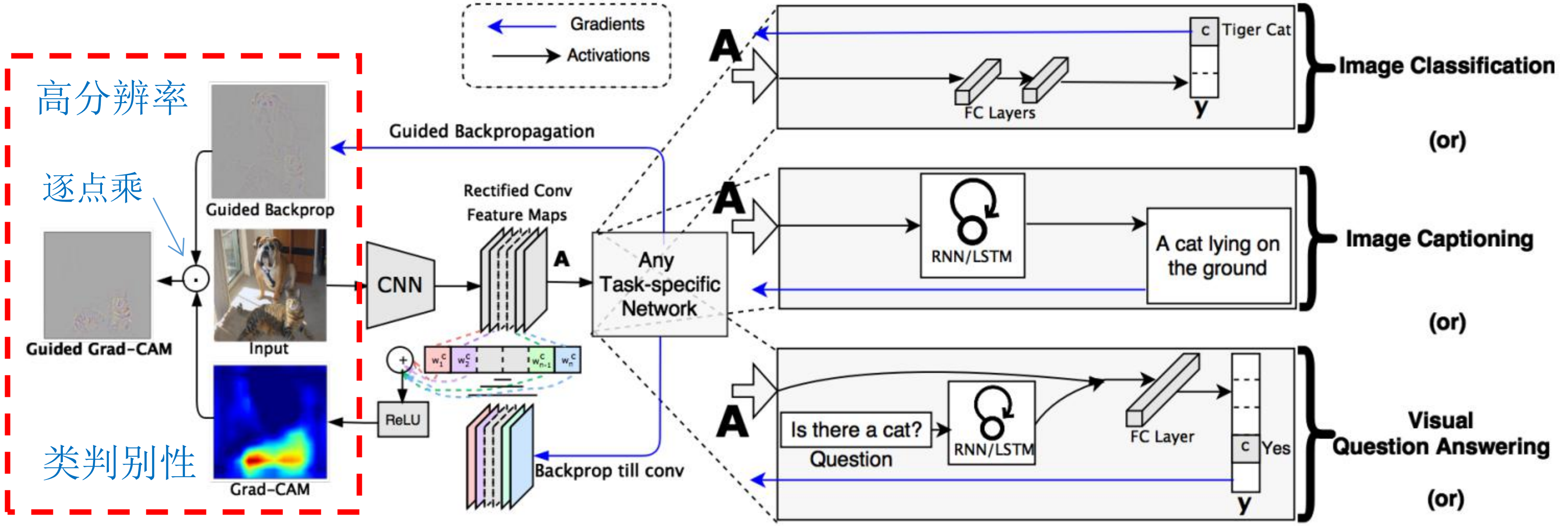
$$L^c_{\text{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha^c_k A^k\right)}_{\text{linear combination}}$$

$$\alpha^c_k = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backprop}}$$

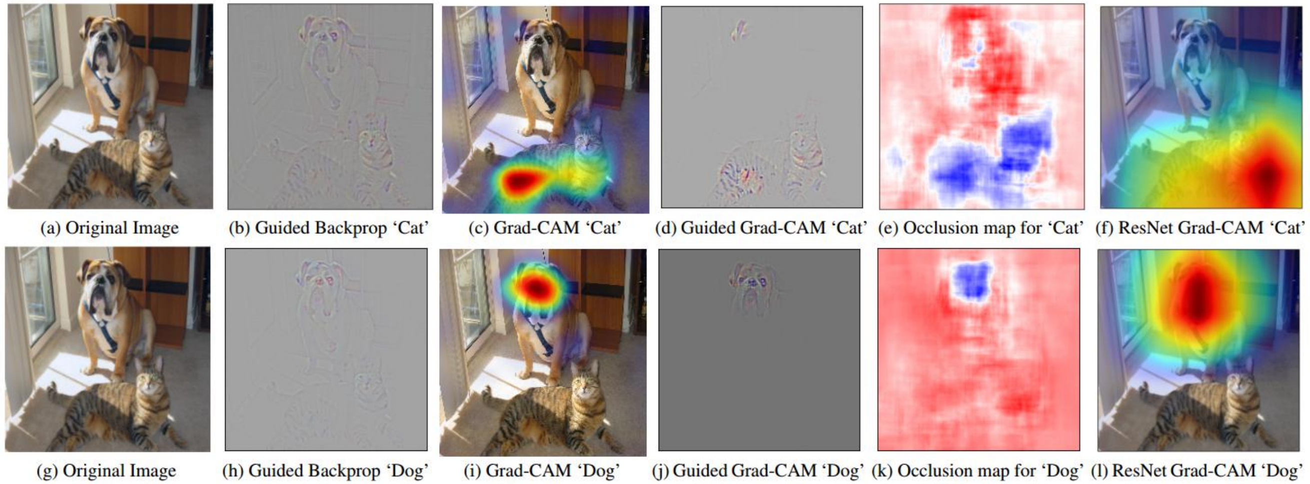
Grad-CAM摆脱了对GAP型特定网络架构的要求

WE HAVE $w^c_k = \sum_i \sum_j \frac{\partial Y^c}{\partial A^k_{ij}}$ (a_k和w_k呈比例关系, 不影响可视化)

(Proof omitted here)



Guide Grad-CAM



(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'

GradCAM++2018

存疑：1.对于分类，遗漏目标是否有影响
2. 假设一个特征图捕捉一个目标是否OK

arxiv2018, Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

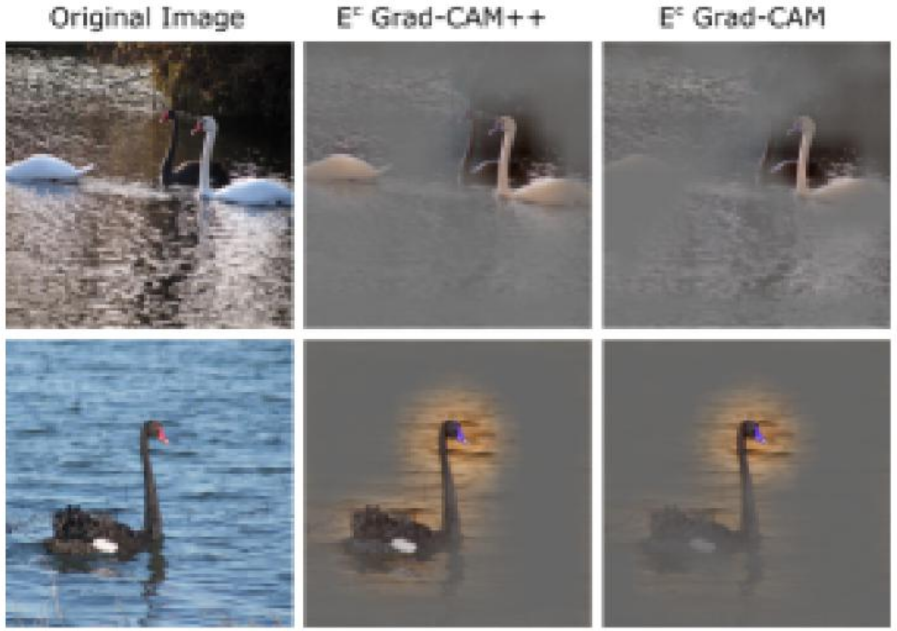


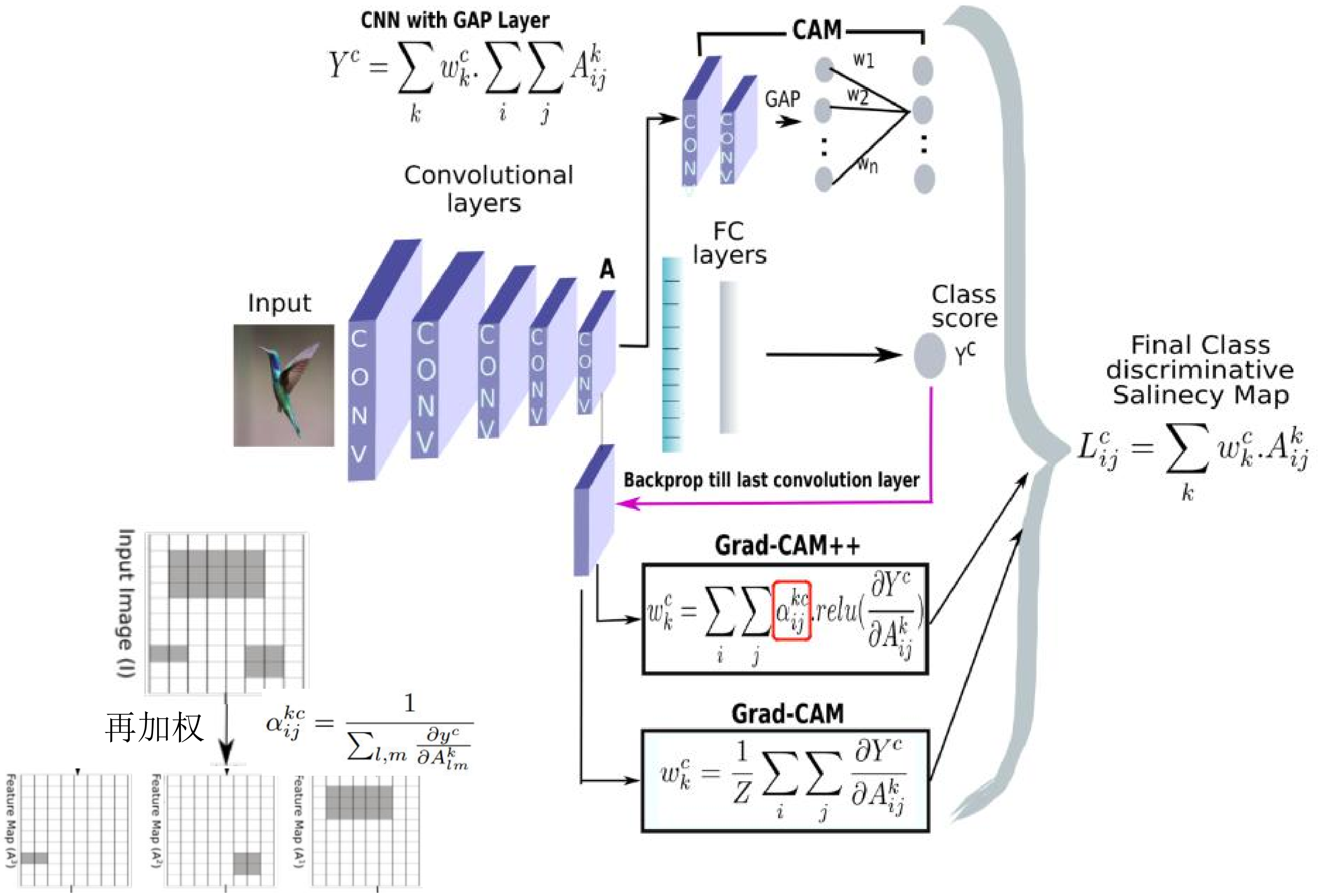
Fig. 4. Example explanation maps for 2 images generated by Grad-CAM++ and Grad-CAM.

- **Two limitations of Grad-CAM**

1. 单图多目标，遗漏目标定位
2. 单图单目标，目标覆盖不全

- **Grad-CAM++ solution**

Grad-CAM的w$_k$定义会使“面积大”(梯度大)的特征图权重大，通过[再加权]去掉面积优势

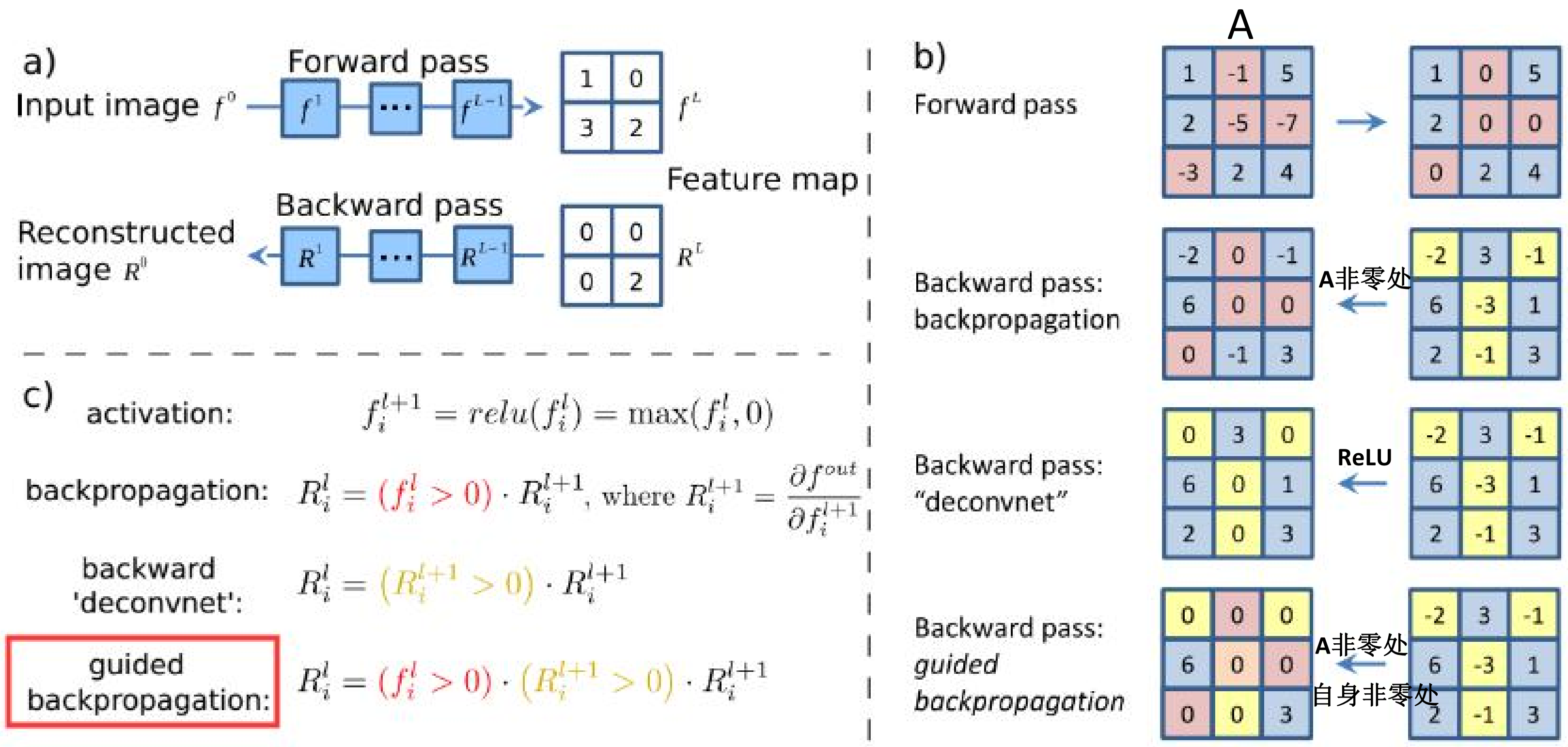


CNN with GAP Layer
$Y^c = \sum_k w_k^c . \sum_i \sum_j A_{ij}^k$
CAM
GAP
w1
w2
wn
CONV
CONV
Convolutional layers
Input
CONV
CONV
CONV
CONV
CONV
A
FC layers
Class score
Y^c
Final Class discriminative Salinecy Map
$L_{ij}^c = \sum_k w_k^c . A_{ij}^k$
Backprop till last convolution layer
Grad-CAM++
$w_k^c = \sum_i \sum_j \boxed{\alpha_{ij}^{kc}} . relu(\frac{\partial Y^c}{\partial A_{ij}^k})$
Grad-CAM
$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$
Input Image (I)
再加权
$\alpha_{ij}^{kc} = \frac{1}{\sum_{l,m} \frac{\partial y^c}{\partial A_{lm}^k}}$
Feature Map (A¹)
Feature Map (A²)
Feature Map (A³)

Fig. 3. An overview of all the three methods – CAM, Grad-CAM, Grad-CAM++ – with their respective computation expressions.

GuidedBackprop2015

ICLR2015, Striving for Simplicity: The All Convolutional Net



1	0
3	2

0	0
0	2

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

b)

Forward pass

A

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

→

1	0	5
2	0	0
0	2	4

Backward pass: backpropagation

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

A非零处 ←

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

Backward pass: “deconvnet”

0	3	0
6	0	1
2	0	3

ReLU ←

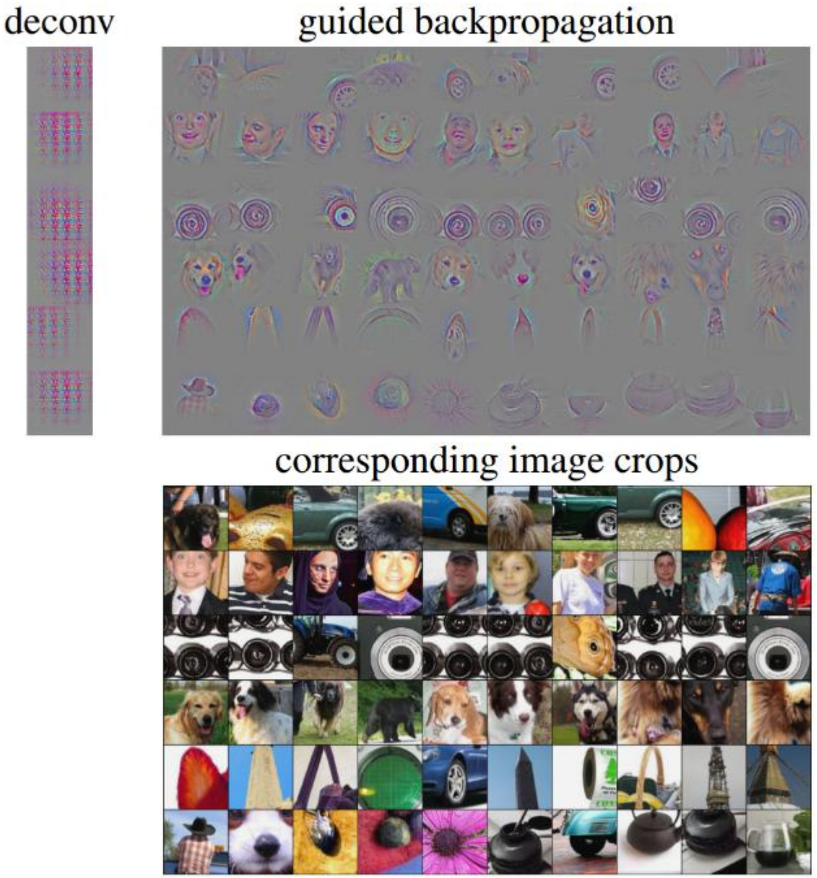
-2	3	-1
6	-3	1
2	-1	3

Backward pass: *guided backpropagation*

0	0	0
6	0	0
0	0	3

A非零处 ← 自身非零处

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



- 上图含义，对于ReLU函数：

1. 对于Backprop，回传梯度值取决于l层自身正负性
2. 对于Deconv，l层的重构特征取决于(l+1)层正负性
3. 对于Guided-Backprop，同时取决于两者

其他: 本文(2015年)尝试全部用3*3conv，尝试用s=2的conv替代maxpooling，也算是早期探索者

Invert2015

CVPR2015, Undrstanding Deep Image Representations by Inverting Them

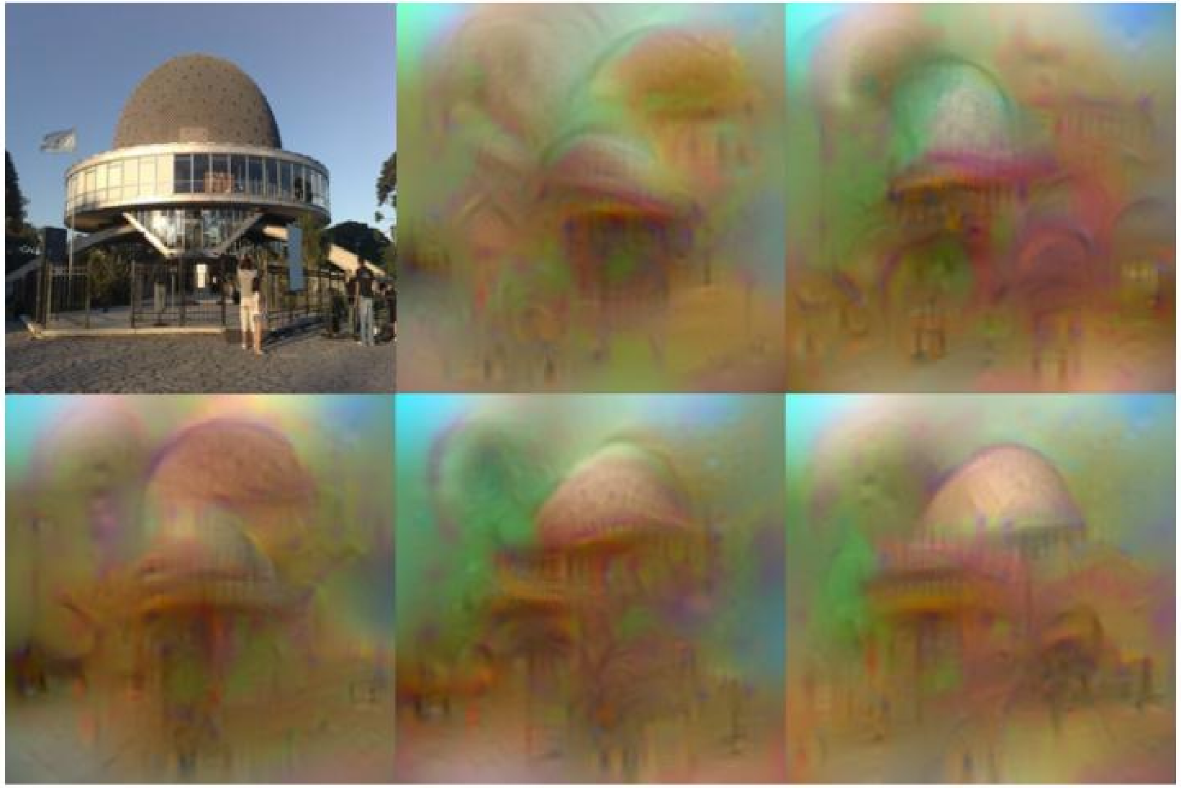


Figure 1. **What is encoded by a CNN?** The figure shows five possible reconstructions of the reference image obtained from the 1,000-dimensional code extracted at the penultimate layer of a reference CNN[13] (before the softmax is applied) trained on the ImageNet data. From the viewpoint of the model, all these images are practically equivalent. This image is best viewed in color/screen.

- **Inverting Pepresentations**: Generating image x that has common feature with original image x_0.

$$\boxed{\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^{H\times W\times C}} \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda\mathcal{R}(\mathbf{x})}$$

1. 在特征空间中约束
2. 解不唯一

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

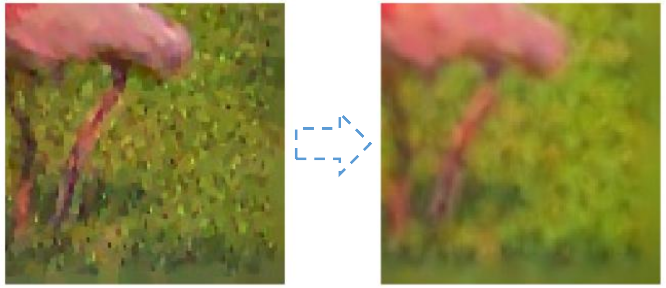
$$\mathcal{R}_\alpha(\mathbf{x}) = \|\mathbf{x}\|_\alpha^\alpha$$

对TV(total variation) norm Rv，不严谨理解为：
水平梯度^2+垂直梯度^2 → 实际梯度^2 ？！

$$\mathcal{R}_{V^\beta}(\mathbf{x}) = \sum_{i,j}\left(\underline{(x_{i,j+1} - x_{ij})^2 + (x_{i+1,j} - x_{ij})^2}\right)^{\frac{\beta}{2}}$$

$$\boxed{\|\Phi(\sigma\mathbf{x}) - \Phi_0\|_2^2/\|\Phi_0\|_2^2 + \lambda_\alpha\mathcal{R}_\alpha(\mathbf{x}) + \lambda_{V^\beta}\mathcal{R}_{V^\beta}(\mathbf{x})}$$

- **Comparison**: DeConvNet(ECCV14) looks at **how** certain network outputs are obtained, while this work looks for **what** information is preserved by the network output.



TV norm效果展示
图为参数β由1增为2

Invert2016

对比：
1. CVPR15：特征空间约束+手动先验正则
2. DeConvNet2014，upconv非训练

CVPR2016, Inverting Visual Representations with Convolutional Networks

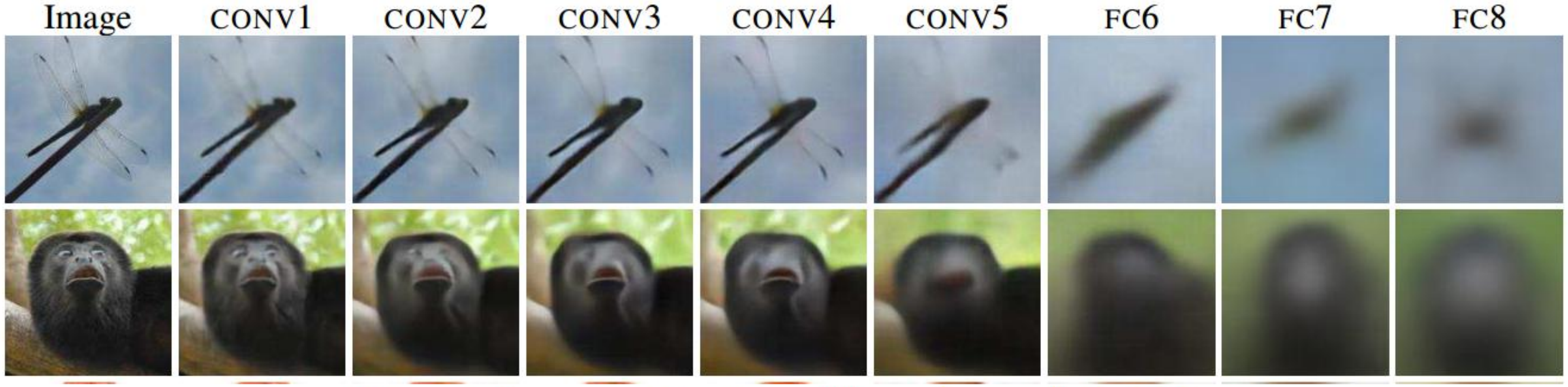
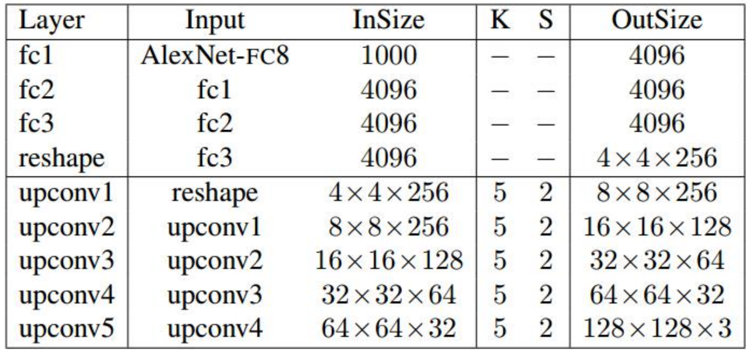


Figure 5: Reconstructions from different layers of AlexNet.

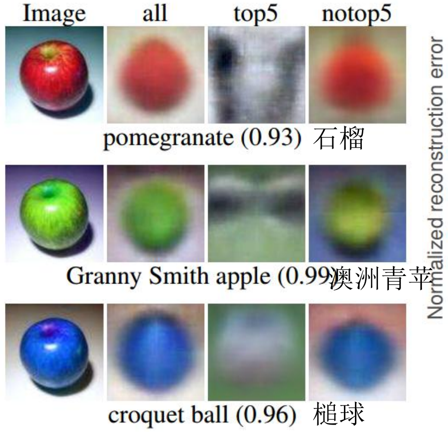


Layer	Input	InSize	K	S	OutSize
fc1	AlexNet-FC8	1000	–	–	4096
fc2	fc1	4096	–	–	4096
fc3	fc2	4096	–	–	4096
reshape	fc3	4096	–	–	4×4×256
upconv1	reshape	4×4×256	5	2	8×8×256
upconv2	upconv1	8×8×256	5	2	16×16×128
upconv3	upconv2	16×16×128	5	2	32×32×64
upconv4	upconv3	32×32×64	5	2	64×64×32
upconv5	upconv4	64×64×32	5	2	128×128×3

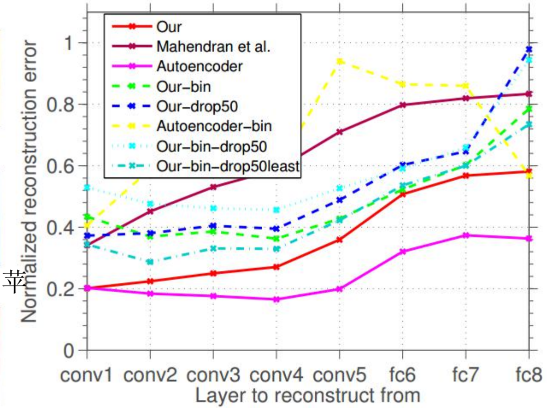
Table 1: Network for reconstructing from AlexNet FC8 features. K stands for kernel size, S for stride.

$$\begin{array}{|c|c|}\hline 25 & 33 \\ \hline 14 & 8 \\ \hline\end{array} \xrightarrow[2x]{上采样} \begin{array}{|c|c|c|c|}\hline 25 & 0 & 33 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 14 & 0 & 8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline\end{array}$$

有点奇怪的upconv？



颜色对分类的影响



不同层的重构误差, bin和drop是指对特征的扰动实验

- **Method**

$$\hat{f}(\phi_0) = \mathbb{E}_{\mathbf{x}}\left[\mathbf{x} \,|\, \phi = \phi_0\right]$$

expected pre-image

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_i ||\mathbf{x}_i - f(\phi_i, \mathbf{w})||_2^2$$

- **Key points**

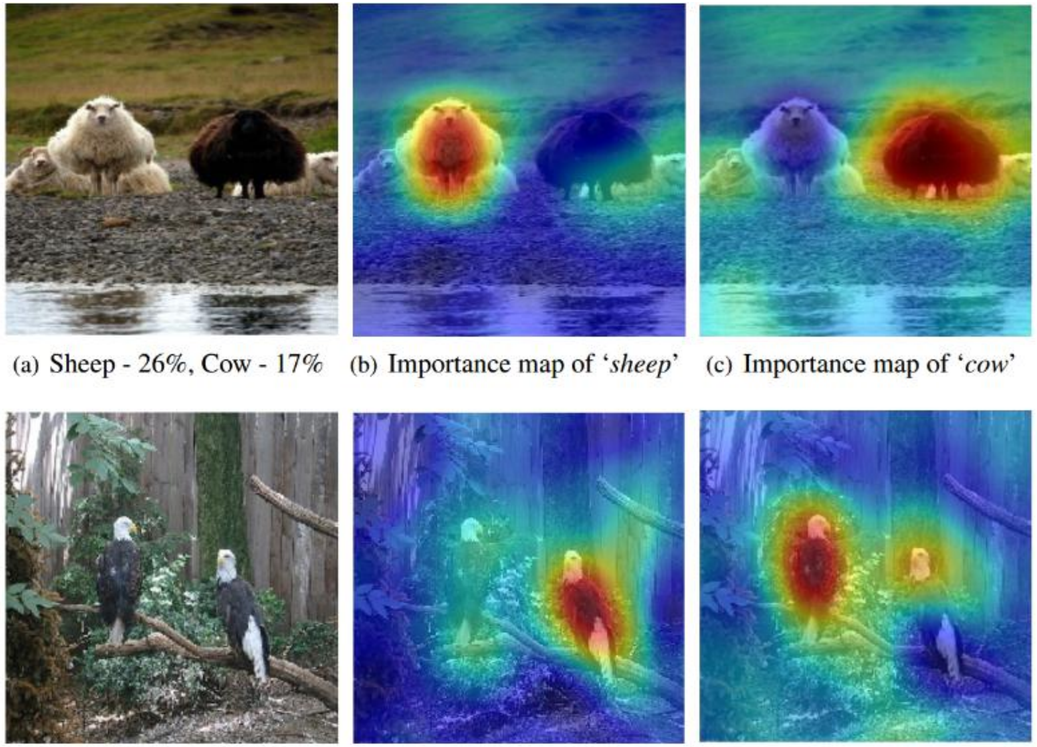
1. 在图像空间中约束
2. 目标函数隐式学习图像先验正则
3. 训练耗时，测试只有前向推理
4. 须专门定义upconv网络

- Experimental Conclusions

1. 各层特征不同程度保留了颜色和位置信息（后面应用会再讨论）
2. 信息在于特征的非零元，具体取值影响不大，故dropout影响大于bin
3. **dark knowledge**: small probs of non-predicted classes carry more info.

RISE2018

arxiv2018, RISE: Randomized Input Sampling for Explanation of Black-box Models



(a) Sheep - 26%, Cow - 17% (b) Importance map of '*sheep*' (c) Importance map of '*cow*'

(d) Bird - 100%, Person - 39% (e) Importance map of '*bird*' (f) Importance map of '*person*'

Figure 1: Our proposed RISE approach can explain why a black-box model (here, ResNet50) makes classification decisions by generating a pixel importance map for each decision (redder is more important). For the top image, it reveals that the model only recognizes the white sheep and confuses the black one with a cow; for the bottom image it confuses parts of birds with a person. (Images taken from the PASCAL VOC dataset.)

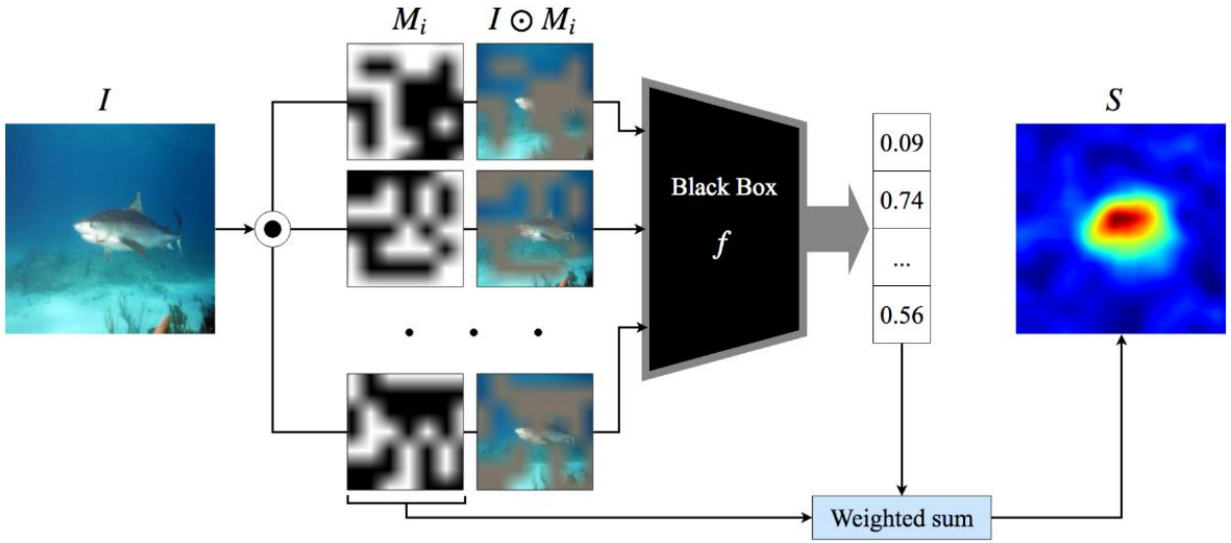
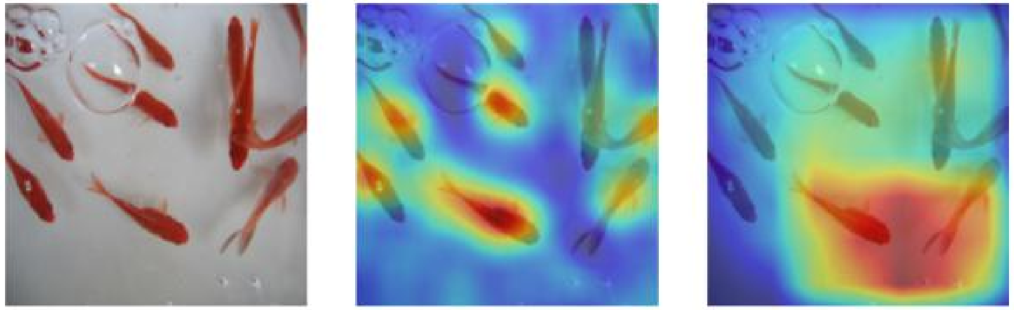


Figure 3: Overview of RISE: Input image I is element-wise multiplied with random masks M_i and the masked images are fed to the base model. The saliency map is a linear combination of the masks where the weights come from the score of the target class corresponding to the respective masked inputs.



(a) Input (b) RISE (ours) (c) GradCAM

Applications

Applications

Architecture Selection

From DeConvNet2014

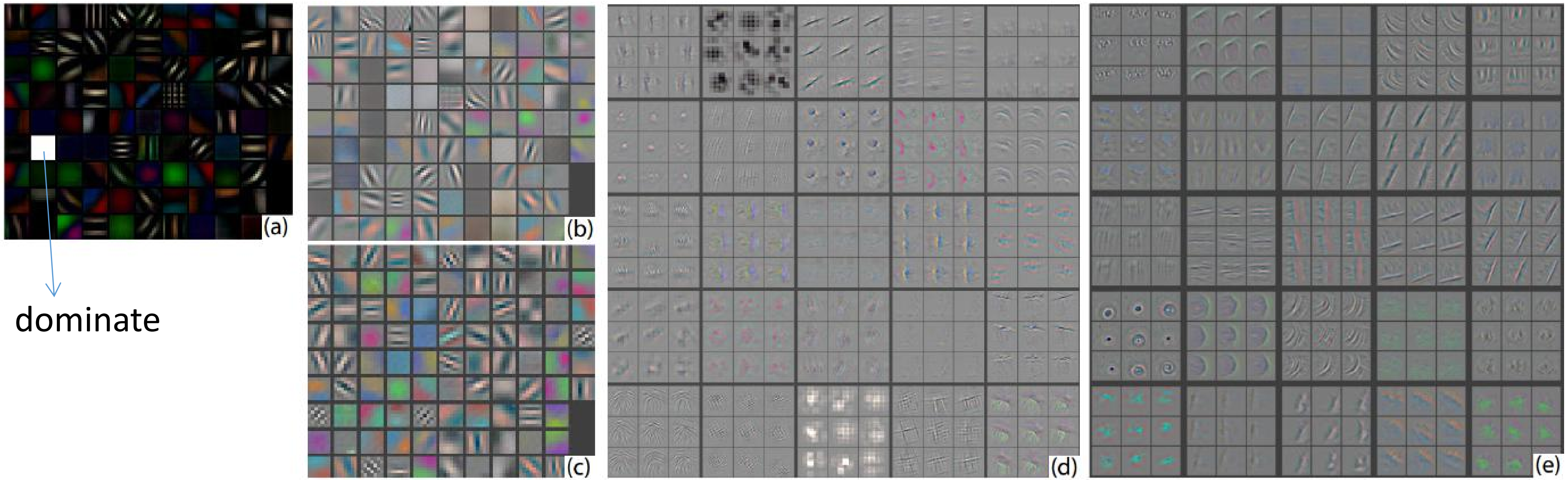


Figure 6. (a): 1st layer features without feature scale clipping. Note that one feature dominates. (b): 1st layer features from (Krizhevsky et al., 2012). (c): Our 1st layer features. The smaller stride (2 vs 4) and filter size (7x7 vs 11x11) results in more distinctive features and fewer "dead" features. (d): Visualizations of 2nd layer features from (Krizhevsky et al., 2012). (e): Visualizations of our 2nd layer features. These are cleaner, with no aliasing artifacts that are visible in (d).

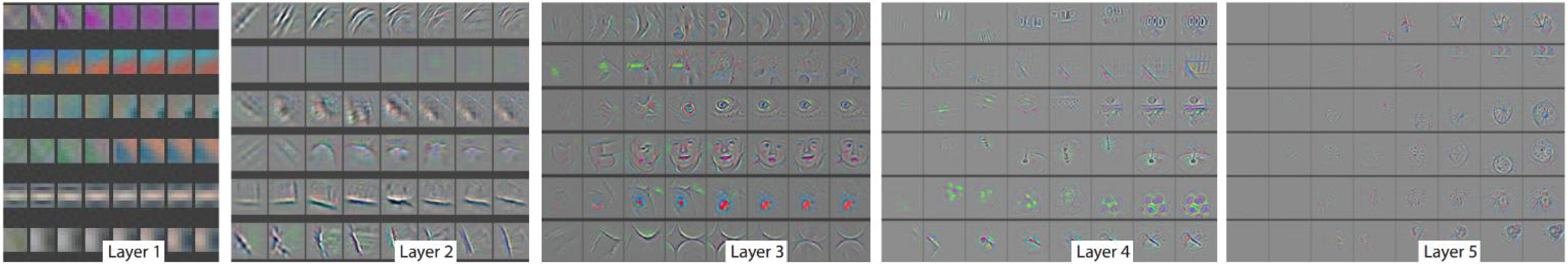
- 现象：第一层(a)主要是高低频信息但少中频信息(?)，有主导特征；第二层(b)有aliasing artifacts
- 措施：第一层换用filter size由11至7、减小stride从4至2，并renormalize filter (RMS > 0.1)；
- 效果：第一层特征图b→c, 第二层特征图d→e

Applications

Network Training Epoches

epoch=0 → epoch=64

From DeConvNet2014



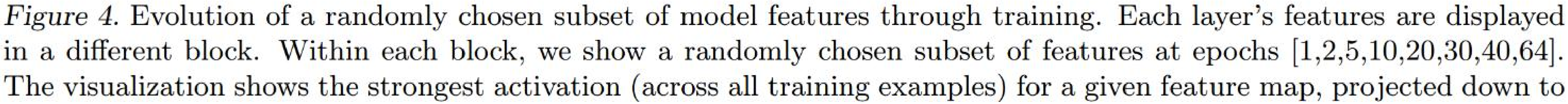


Figure 4. Evolution of a randomly chosen subset of model features through training. Each layer's features are displayed in a different block. Within each block, we show a randomly chosen subset of features at epochs [1,2,5,10,20,30,40,64]. The visualization shows the strongest activation (across all training examples) for a given feature map, projected down to

- 论文结论：横向看，底层特征易训练，高层特征须训练更多epoch
- 个人解释：高层的特征是依赖于底层特征的，自然要在底层训练完毕之后才训练得好
- 其他观点：网络加深，梯度难回传，不应该是底层难训练？！

Applications

补充：随着深度的增加，特征由个体带来的“风格”差异减小，由类别带来的“内容”差异增大。
或者说，随着深度的增加，模型更关注不变性和本质，同类个体间的差异被无视。

Network Understanding

From Invert2016

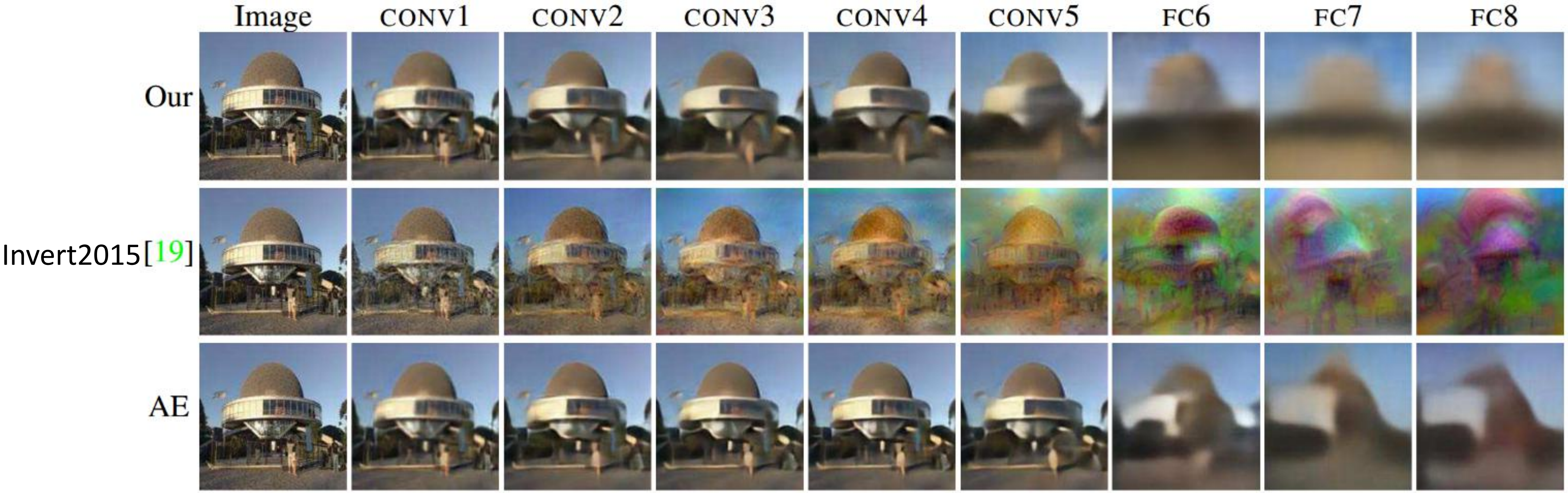


Figure 6: Reconstructions from layers of AlexNet with our method (top), [19] (middle), and autoencoders (bottom)

1st行：若模型足够好，则同类不同照片对应几乎一样的特征，因此在图像空间重构，会用模糊以cover不同样本！另外高层特征细节信息的丢失，会加大模糊~

2nd行：分别输入该行图片，能近似得到输入原始Img得到的特征，因在特征空间重构，所以生成的图像会丢失颜色和位置信息。

- **不变性 VS. 判别性 （个人理解）** → 见补充文档：2-3_VisDNN_关于不变性和判别性的一些思考-熊凯.docx

1. 高层(fc6~fc8)的不变性强于底层，但代价是丢失了很多信息比如目标位置、颜色
2. MaxPooling是为了增加区域不变性，自然也损失了信息，GuidedBackprop2015中就探索了用步长为2的Conv替代MaxPooling，现已在人脸识别等网络中广泛使用
3. 风格迁移中认为：图像特征图各通道的均值和标准差，编码了图像的风格信息，所以InstanceNorm可实现风格不变性，不变性会损害判别性，所以多图分类常用BN，单图迁移多用IN ？！ 一种解释认为均值标准差也编码了判别信息，故IN后被损失 -- 存疑

Applications

Few-Shot Learning亟需推理能力，或许也能从更有效地利用样本开始

Fine-grained Recognition

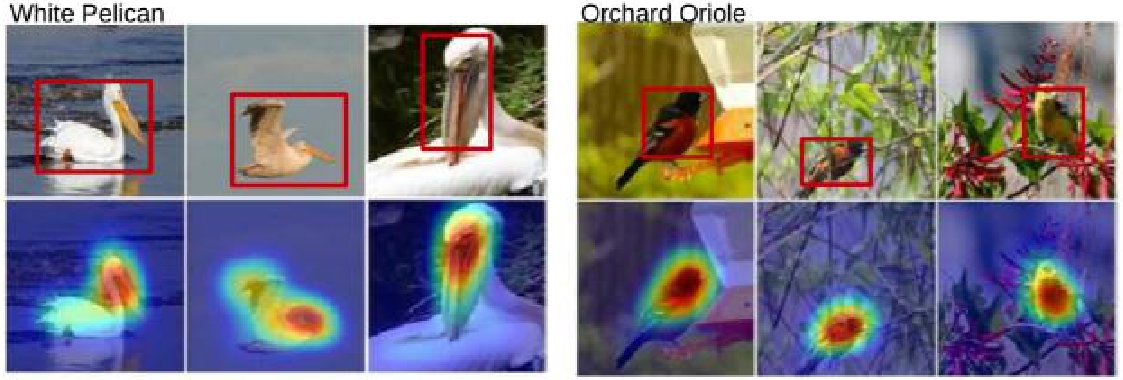
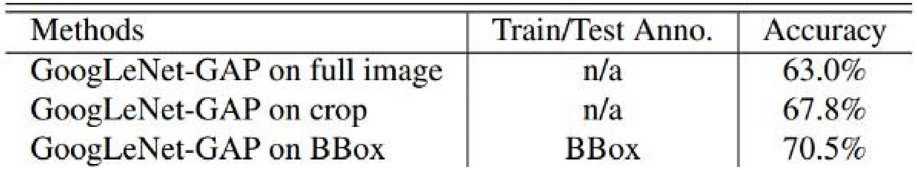


Figure 7. CAMs and the inferred bounding boxes (in red) for selected images from four bird categories in CUB200. In Sec. 4.1 we



Methods	Train/Test Anno.	Accuracy
GoogLeNet-GAP on full image	n/a	63.0%
GoogLeNet-GAP on crop	n/a	67.8%
GoogLeNet-GAP on BBox	BBox	70.5%

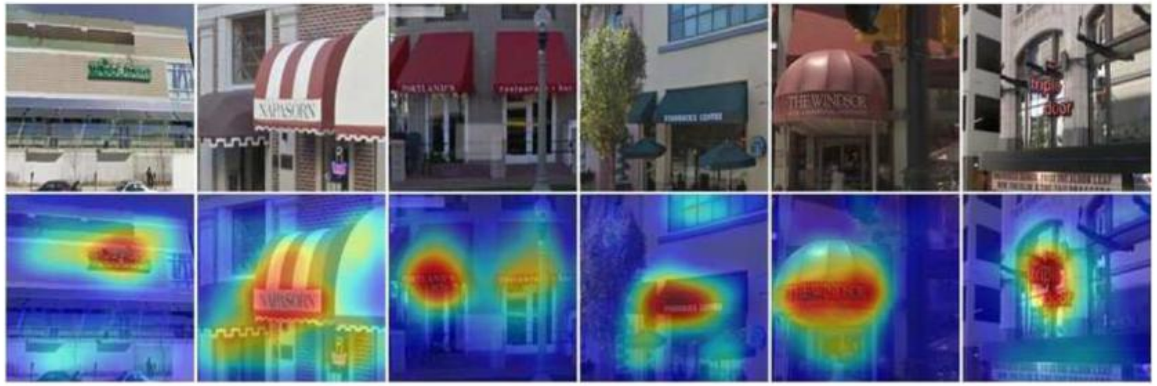


Figure 11. Learning a weakly supervised text detector. The text is accurately detected on the image even though our network is not trained with text or any bounding box annotations. From CAM2016

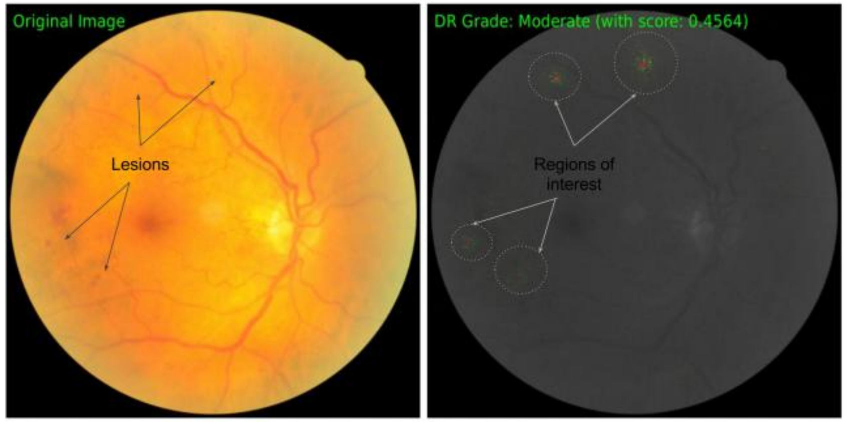


Figure 3. **Attribution for Diabetic Retinopathy grade prediction from a retinal fundus image.** The original image is show on the left, and the attributions (overlayed on the original image in gray scaee) is shown on the right. On the original image we annotate lesions visible to a human, and confirm that the attributions indeed point to them.

From ICLR2017-Axiomatic Attribution for Deep Networks

Applications

From DeepInside2014

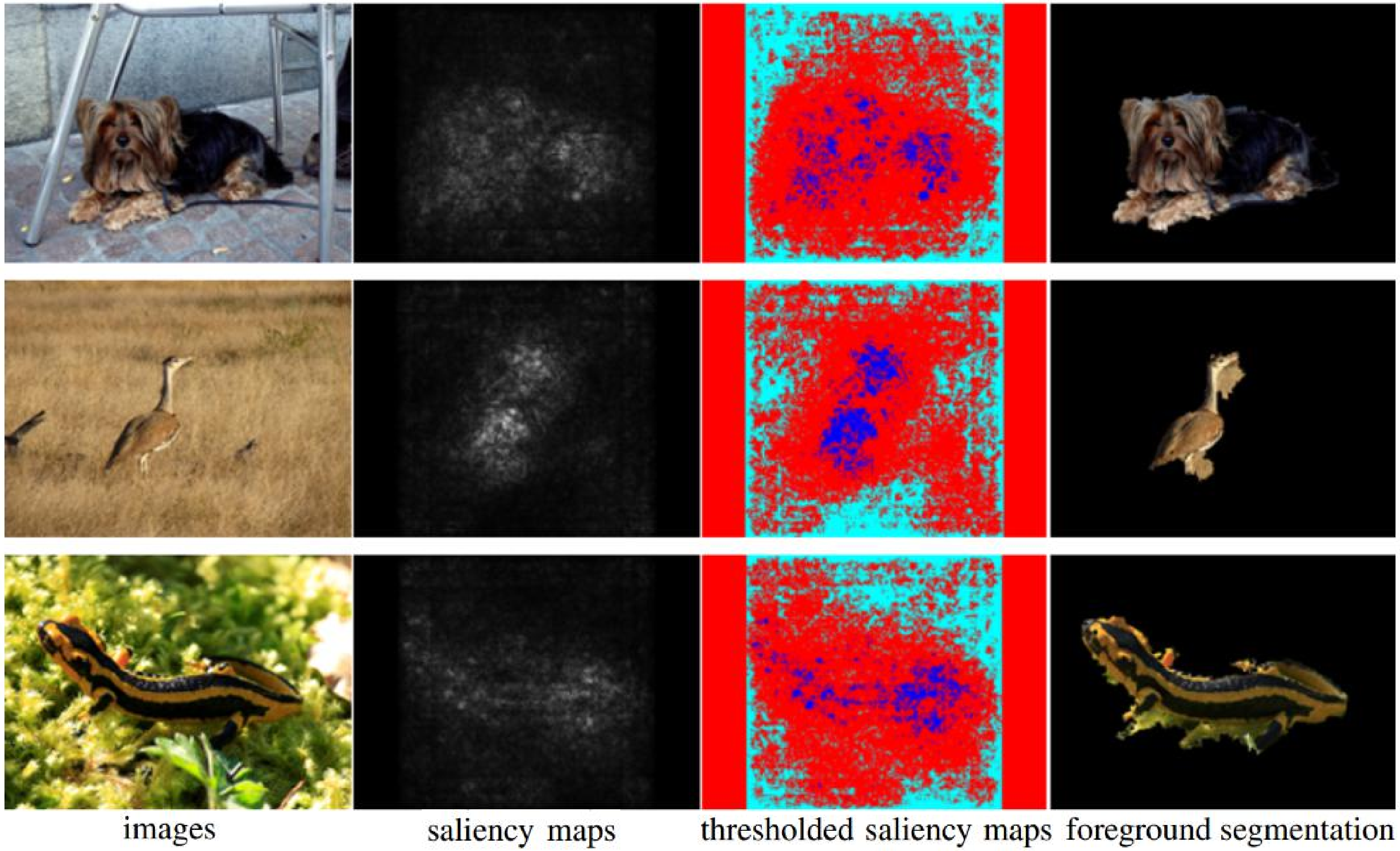


Figure 3: **Weakly supervised object segmentation using ConvNets (Sect. 3.2).**

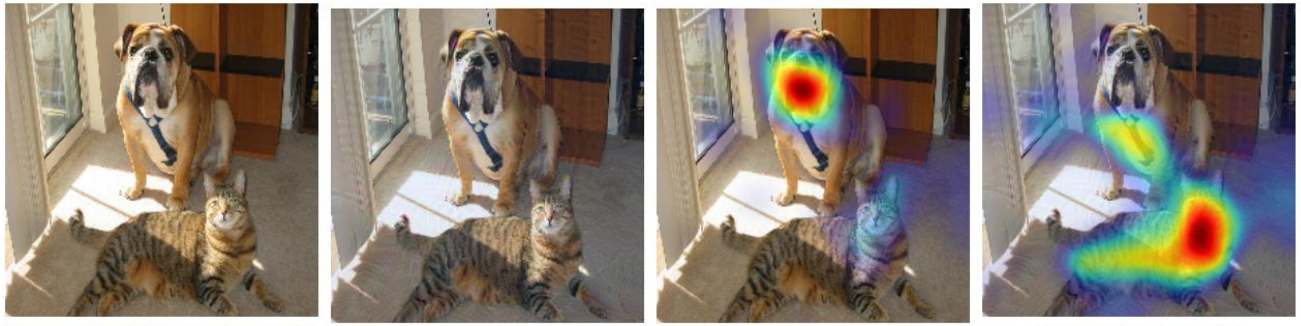
1. Foreground and background colour models: Gaussian Mixture Models
2. GraphCut[1] colour segmentation

[1] http://www.robots.ox.ac.uk/~vgg/software/iseg/

Applications From Grad-CAM

Interesting: 对抗样本改变了最终的预测结果，但实际并未改变网络的注意力，所以到底是什么造成结果的改变

Effect of adversarial noise on VGG-16

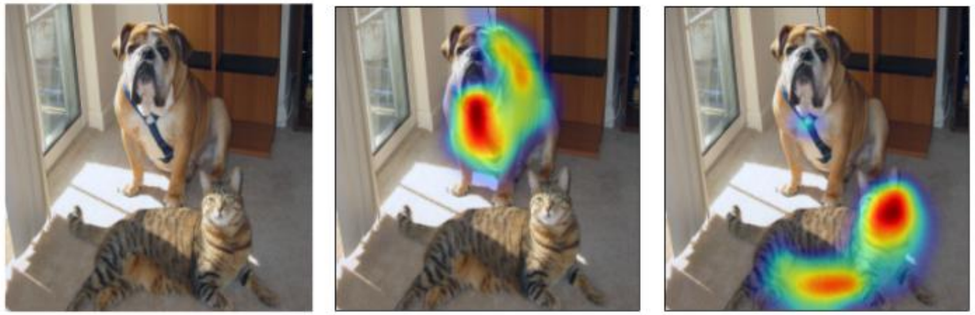


Boxer: 0.40 Tiger Cat: 0.18 Airliner: 0.9999 Boxer: 1.1e-20 Tiger Cat: 6.5e-17

(a) Original image (b) Adversarial image (c) Grad-CAM “Dog” (d) Grad-CAM “Cat”

Figure 5: (a-b) Original image and the generated adversarial image for category “airliner”. (c-d) Grad-CAM visualizations for the original categories “tiger cat” and “boxer (dog)” along with their confidence. Inspite of the network being completely fooled into thinking that the image belongs to “airliner” category with high confidence (>0.9999), Grad-CAM can localize the original categories accurately.

Counterfactual Explanations



(a) Original Image (b) Cat Counterfactual exp (c) Dog Counterfactual exp

Figure 6: Negative Explanations with Grad-CAM

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

6.1. Analyzing Failure Modes for VGG-16

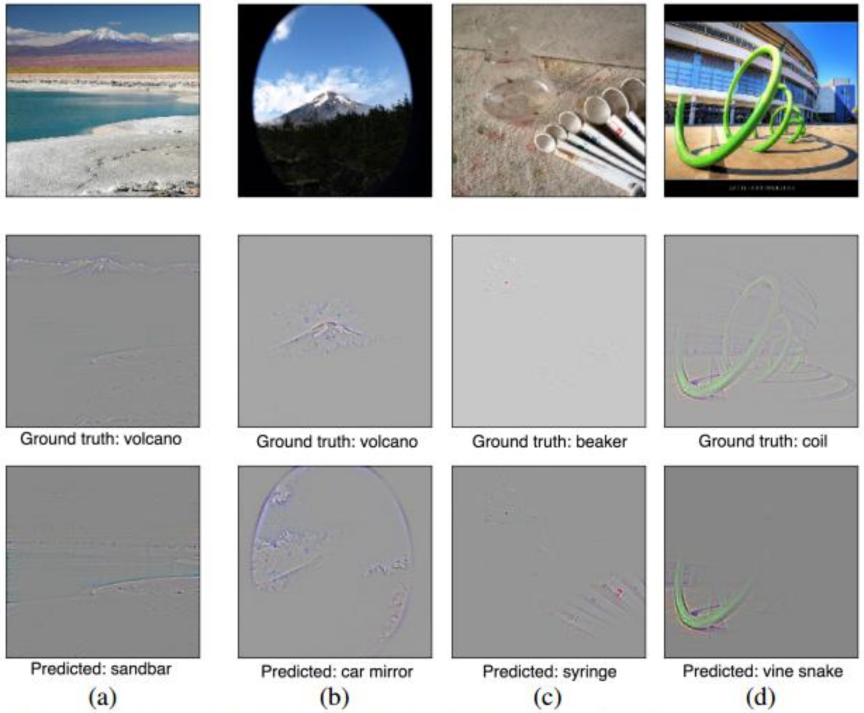
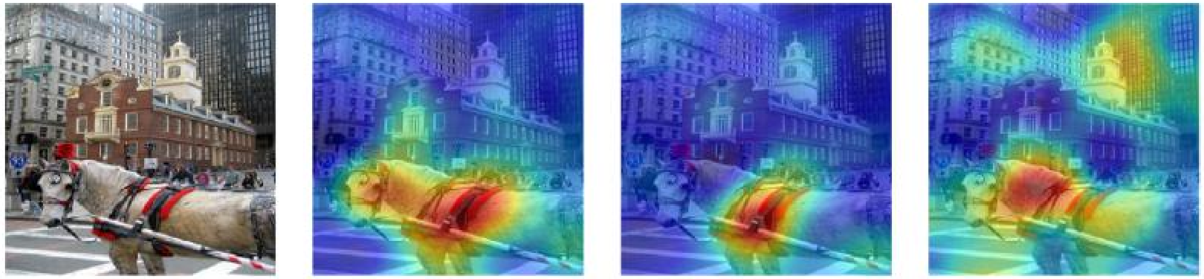


Figure 4: In these cases the model (VGG-16) failed to predict the correct class in its top 1 (a and d) and top 5 (b and c) predictions. Humans would find it hard to explain some of these predictions without looking at the visualization for the predicted class.

6.3. Identifying bias in dataset

eg. 医生护士二分类问题，模型泛化性不好，可视化发现关注的是脸部和发型，因此结果gender-biased，对应平衡数据集中的样本分布解决了问题

Applications



(a) "A horse and carriage on a city street."

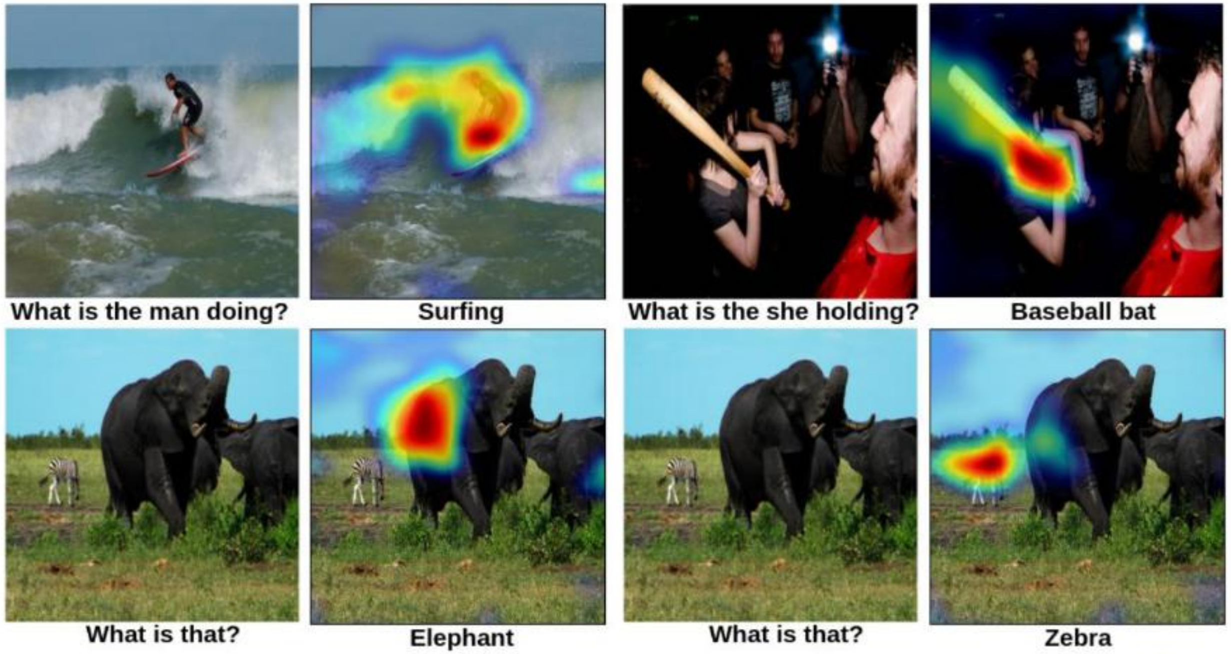
(b) "A horse..."

(c) "A horse and carriage..."

(d) "White..."

Figure 5: Explanations of image captioning models. From RISE2018

Visualize Image Caption & VQA



What is the man doing?

Surfing

What is the she holding?

Baseball bat

What is that?

Elephant

What is that?

Zebra

(b) Visualizing ResNet based Hierarchical co-attention VQA model from [33]

From Grad-CAM2017

Resources

Convolutional Neural Network Visualizations

This repository contains a number of convolutional neural network visualization techniques implemented in PyTorch.

https://github.com/utkuozbulak/pytorch-cnn-visualizations/tree/master/src

- **Model Visualization**

1. pytorchviz: https://github.com/szagoruyko/pytorchviz

2. torchsummary: https://github.com/sksq96/pytorch-summary

3. graphviz + torchviz

pip install graphviz

pip install git+https://github.com/szagoruyko/pytorchviz

4. tensorboardX: https://github.com/lanpa/tensorboardX

5. tensorwatch + jupyter notebook: https://github.com/microsoft/tensorwatch

（含Lime: https://github.com/marcotcr/lime）

6. netron

https://lutzroeder.github.io/netron/ (网页版)

https://github.com/lutzroeder/netron (github主页)

7. HiddenLayer: https://github.com/waleedka/hiddenlayer

8. visdom: https://github.com/facebookresearch/visdom

9. VisualDL (PaddlePaddle): https://github.com/PaddlePaddle/visualdl

- 模型可视化个人小结

1. 优先试试网页版的netron
2. 使用经典的tensorboardX（可能版本不兼容报错）
3. 使用tensorwatch(+jupyter notebook)

The End

Thanks & Questions