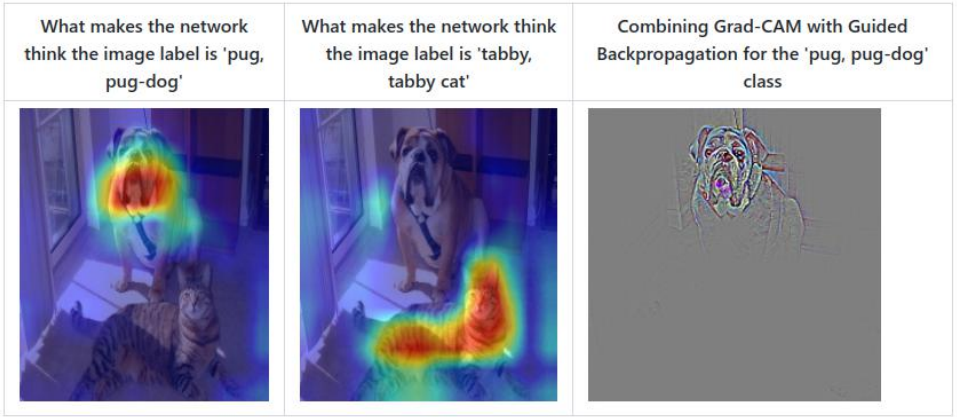
Grad-CAM

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

ICCV'17

Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra



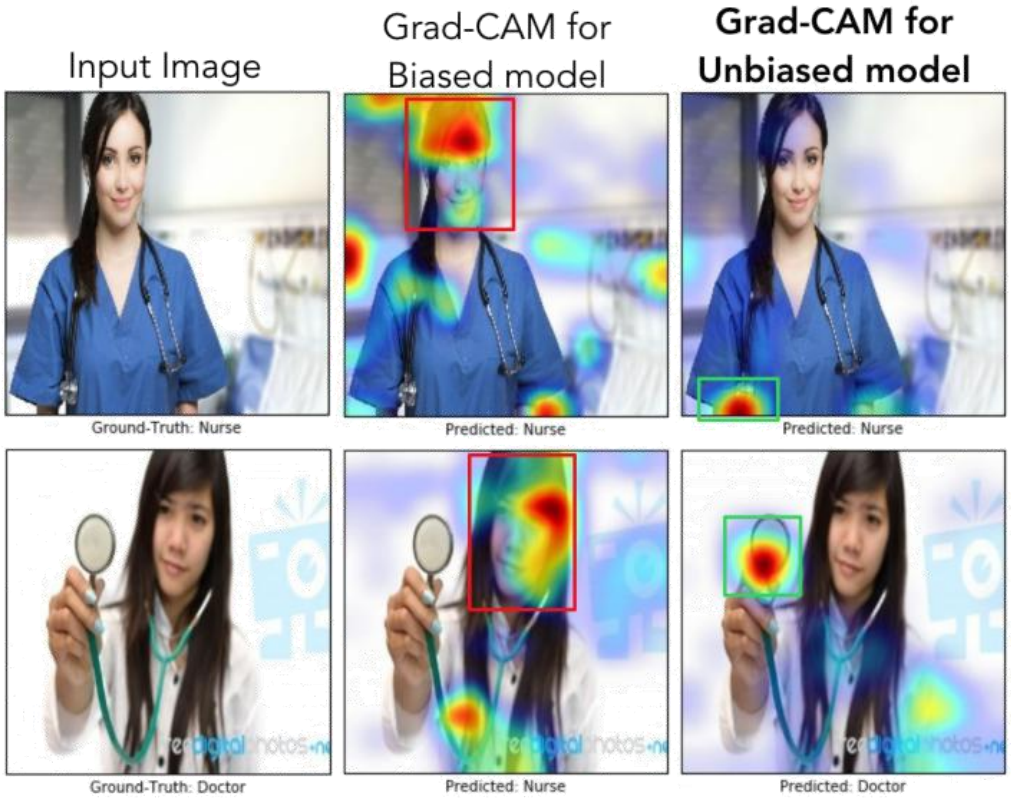
CAM
(Class Activation Mapping)

Grad-CAM
(Gradient-weighted Class Activation Mapping)

论文下载地址：https://arxiv.org/abs/1610.02391
推荐博文：https://blog.csdn.net/qq_37541097/article/details/123089851
推荐代码（Pytorch）：https://github.com/jacobgil/pytorch-grad-cam

Grad-CAM

6.3 Identifying bias in dataset



Grad-CAM

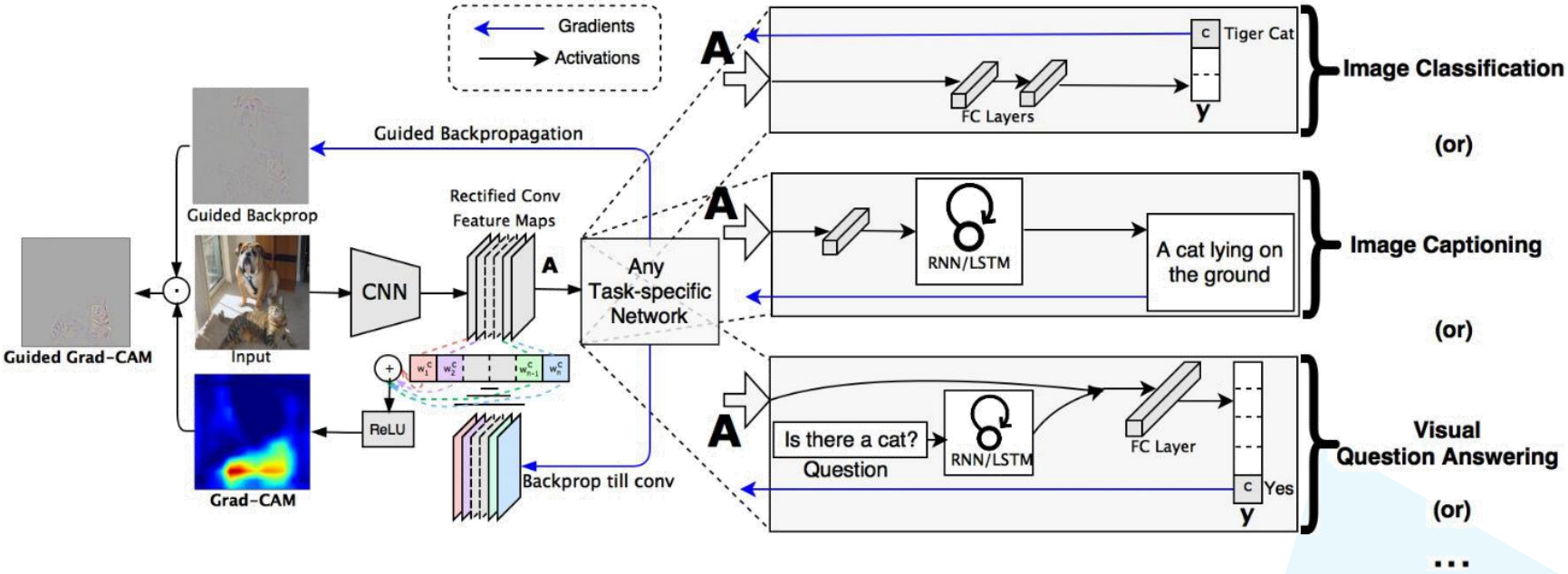


Fig. 2: Grad-CAM overview: Given an image and a class of interest (*e.g.*, 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

Grad-CAM

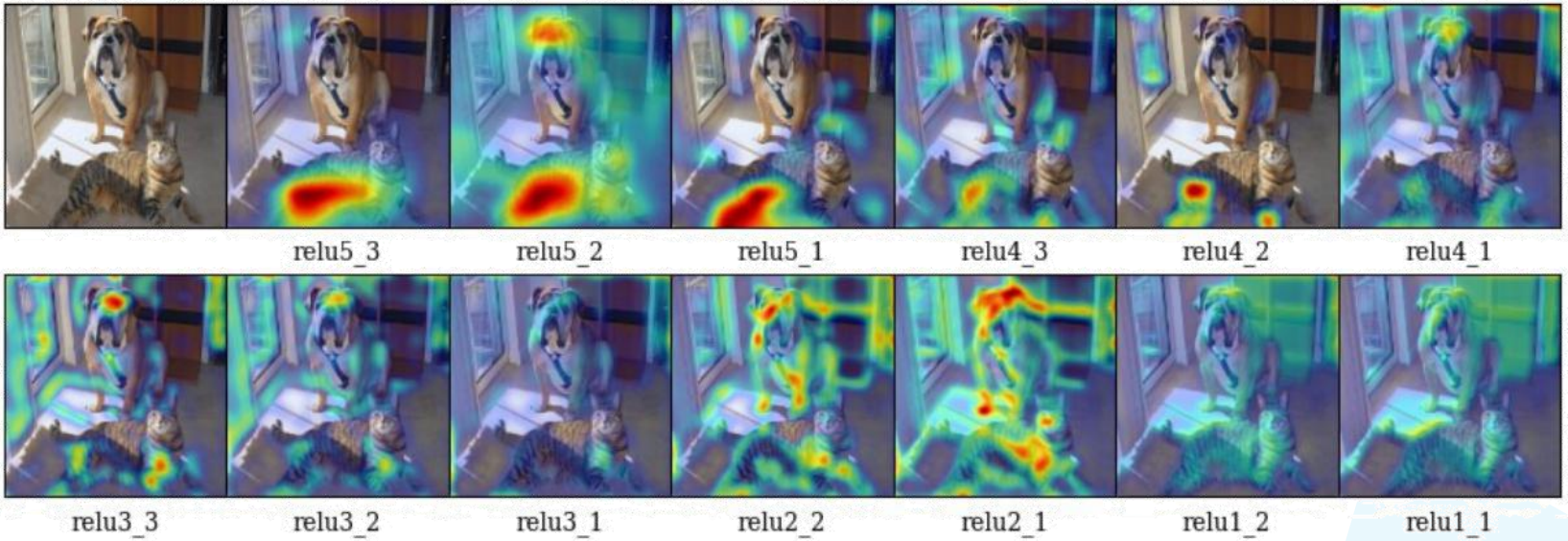


Fig. 13: Grad-CAM at different convolutional layers for the 'tiger cat' class. This figure analyzes how localizations change qualitatively as we perform Grad-CAM with respect to different feature maps in a CNN (VGG16 [52]). We find that the best looking visualizations are often obtained after the deepest convolutional layer in the network, and localizations get progressively worse at shallower layers. This is consistent with our intuition described in Section 3 of main paper, that deeper convolutional layer capture more semantic concepts.

Grad-CAM

$$L^c_{\text{Grad-CAM}} = ReLU(\sum_k \alpha_k^c A^k) \qquad (1)$$

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \qquad (2)$$

- A代表某个特征层（一般取最后一个卷积层的输出）
- k 代表特征层A中第k个通道（channel）
- c 代表关注的类别c
- A^k 代表特征层A中第k个通道的数据
- α_k^c 代表对于类别c，特征层A第k个通道的权重

- y^c 代表网络针对类别c预测的分数(score)，注意这里没有通过softmax激活
- A_{ij}^k 代表特征层A在通道k中，坐标为ij位置处的数据
- Z 代表特征层的宽度乘以高度

Forward

CNN Extractor

1	0	2
3	5	0
1	1	1

$k = 1$

0	1	0
3	1	0
1	0	1

$k = 2$

A

GAP

FC

0.3
0.2
⋮
0
0

Cat

Dog

0	1	0
0	0	0
1	0	1

$k = 1$

0	-1	-1
-1	-1	0
0	-1	-1

$k = 2$

$\acute{A}$

Backward

Grad-CAM

A

1	0	2
3	5	0
1	1	1

$k = 1$

0	1	0
3	1	0
1	0	1

$k = 2$

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \qquad (2)$$

$\acute{A}$

0	1	0
0	0	0
1	0	1

$k = 1$

0	-1	-1
-1	-1	0
0	-1	-1

$k = 2$

$$\alpha^{\text{Cat}} = \begin{pmatrix} \alpha_1^{\text{Cat}} \\ \\ \alpha_2^{\text{Cat}} \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \\ -\frac{2}{3} \end{pmatrix}$$

Grad-CAM

A

1	0	2
3	5	0
1	1	1

$k = 1$

0	1	0
3	1	0
1	0	1

$k = 2$

$\acute{A}$

0	1	0
0	0	0
1	0	1

$k = 1$

0	-1	-1
-1	-1	0
0	-1	-1

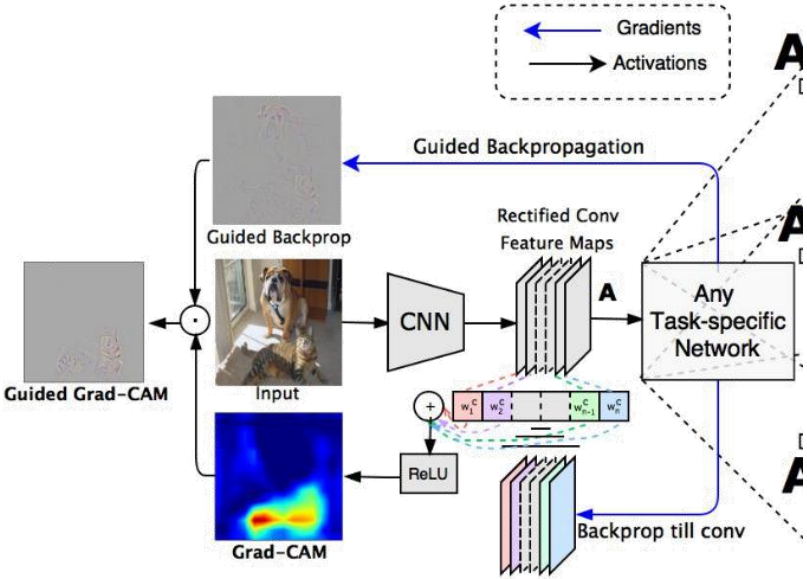
$k = 2$

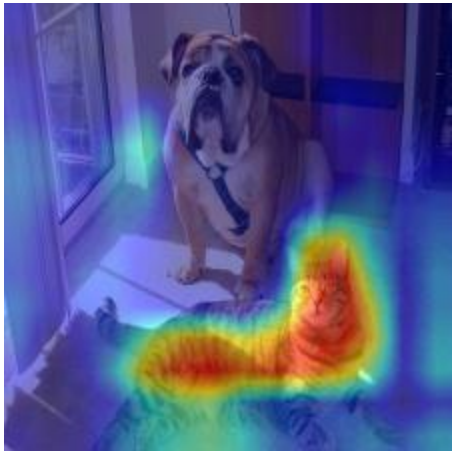
$$L^c_{\text{Grad-CAM}} = ReLU(\sum_k \alpha^c_k A^k) \qquad (1)$$

$$\alpha^{\text{Cat}} = \begin{pmatrix} \alpha^{\text{Cat}}_1 \\ \\ \alpha^{\text{Cat}}_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \\ -\frac{2}{3} \end{pmatrix}$$

$$\begin{aligned} L^{\text{Cat}}_{\text{Grad-CAM}} &= ReLU(\frac{1}{3} \cdot \begin{pmatrix} 1 & 0 & 2 \\ 3 & 5 & 0 \\ 1 & 1 & 1 \end{pmatrix} + (-\frac{2}{3}) \cdot \begin{pmatrix} 0 & 1 & 0 \\ 3 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}) \\ &= ReLU(\begin{pmatrix} \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ -1 & 1 & 0 \\ -\frac{1}{3} & \frac{1}{3} & -\frac{1}{3} \end{pmatrix}) \\ &= \begin{pmatrix} \frac{1}{3} & 0 & \frac{2}{3} \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & 0 \end{pmatrix} \end{aligned}$$

Grad-CAM


Gradients
Activations
Guided Backpropagation
Guided Backprop
Guided Grad-CAM
Input
CNN
Rectified Conv
Feature Maps
A
Any
Task-specific
Network
ReLU
Grad-CAM
Backprop till conv



Grad-CAM

$$x = x - \min(x)$$

$$x = \frac{x}{\max(x)}$$

Scale

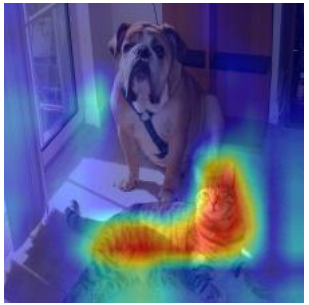
Resize

Color

Scale

Add

Scale



$$img = \frac{img}{255}$$

$$cam = \frac{cam}{cam_{max}} \cdot 255$$