Grad-CAM

why they predict what they predict

Contents

Introduction

Deep Learning in Computer Vision has achieved breakthrough performance in number of tasks like image classification, object detection etc.

However, it is very difficult to interpret the model and deep CNN remains to be a black box for most of us.

Interpretability of the model builds trust but interpretability should not necessarily comprise accuracy.

Source [1]

Introduction

Useful in three stages of AI:

1. AI is weaker than humans - to identify the failure of models
2. AI is on par with humans - to build trust in users
3. AI is stronger than humans - machine teaching

What makes a good visual explanation?

1. Class discriminative - localize the category in the image
2. High resolution - capture fine grain details

Source [1]

Grad-CAM

Grad-CAM - Gradient-Weighted Class Activation Mapping

A technique to visually explain a decision/prediction made by CNN based models to make it more transparent and explainable.

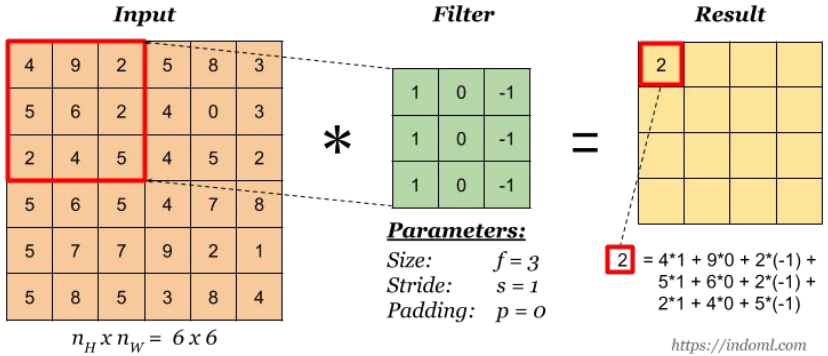
It is a way to understand what part of the images influenced the model to make a decision/prediction.

Grad-CAM is intuitive and simple to implement.

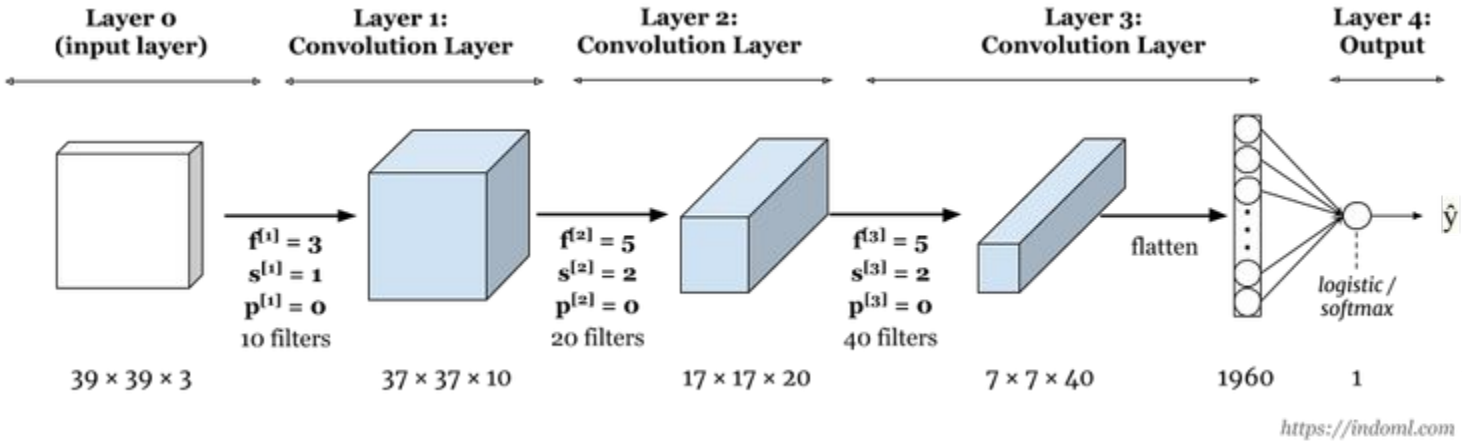
It gives a class discriminative localization map.

Source [1]

CNN Basics



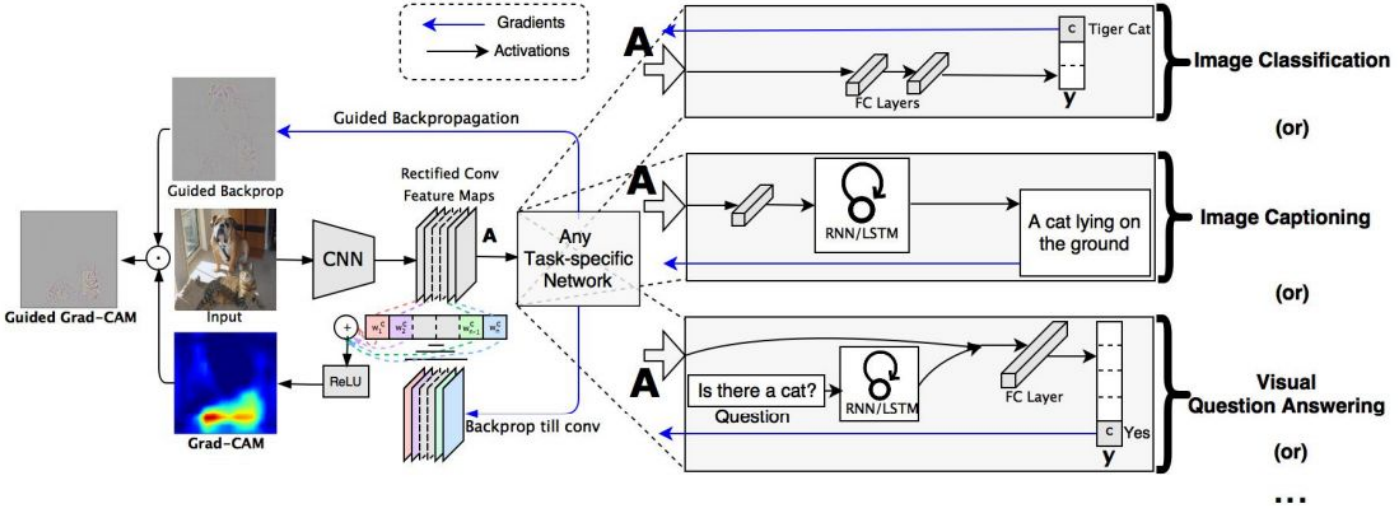
$$n^{[l]} = \left\lfloor \frac{n^{[l-1]} + 2p^{[l-1]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$$



Source [7]

Grad-CAM Overview

Given an image and a class of interest as input, forward propagate the image through the CNN part of the model and then through any task specific computation to obtain a raw score of the category.
The gradient are set to zero for all classes except the desired class which is set to 1.
The signal is then back-propagated to the rectified convolutional feature maps of interest which is then combined to produce Grad-CAM heat map.
Grad-CAM heat map explains where the model has to look to make the particular decision.



Source [1]

Method

Step 1: compute the gradient of the score for class *c*, y^c (before softmax layer) with respect to feature map activations of a convolution layer

$$(\partial y^c)/(\partial A^k)$$

Step 2: global average pooling of the gradient >> neuron importance weights

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

Step 3: weighted combination of activation maps and ReLU

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

Source [1]

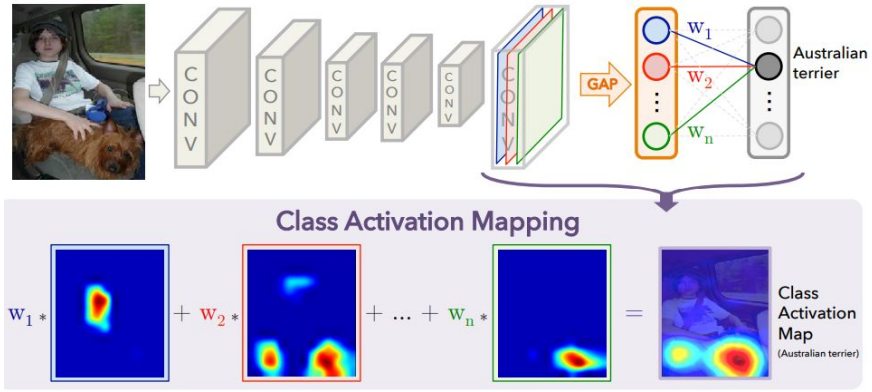
CAM vs. Grad-CAM

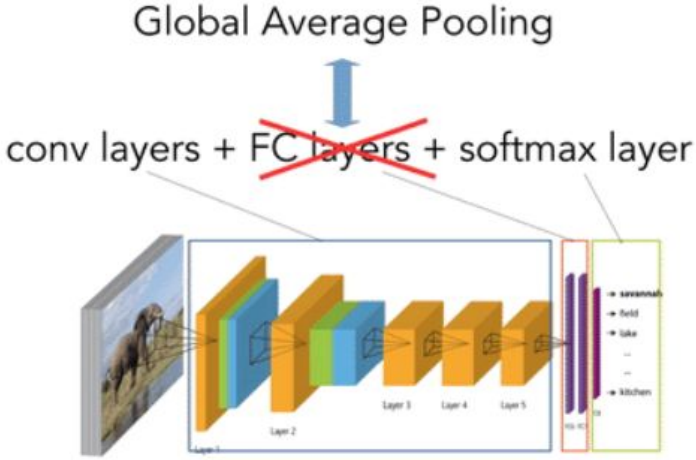
CAM requires a specific architecture. It needs GAP before the softmax layer replacing the FC layer. It comprises model accuracy.

Grad-CAM generalize the CAM for any off the shelf CNN based architectures.

No modification and re-training is required in Grad-CAM.

Grad-CAM is class discriminative and high resolution.





Source [2], [4]

Grad-CAM generalizes CAM

In CAM, the feature maps Ak are spatially pooled using GAP and linearly transformed to produce score Yc

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

Let us define Fk to be the global average pool output,

$$F^k = \frac{1}{Z}\sum_i\sum_j A_{ij}^k \longrightarrow (2)$$

CAM computes the final score by,

$$Y^c = \sum_k w_k^c \cdot F^k \longrightarrow (3)$$

Source [1]

Grad-CAM generalizes CAM

Taking gradient of score Yc with respect to the feature maps Fk

$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A^k_{ij}}}{\frac{\partial F^k}{\partial A^k_{ij}}}$$

Taking partial derivative of eqn.2 w.r.t Ak,

$$\frac{\partial Y^c}{\partial F^k} = \frac{\partial Y^c}{\partial A^k_{ij}} \cdot Z$$

Taking partial derivative of eqn.3 w.r.t Fk,

$$w^c_k = Z \cdot \frac{\partial Y^c}{\partial A^k_{ij}}$$

Source [1]

Grad-CAM generalizes CAM

Sum over all pixels i, j,

$$\sum_i \sum_j w_k^c = \sum_i \sum_j Z \cdot \frac{\partial Y^c}{\partial A_{ij}^k}$$

Re-writing,

$$Z w_k^c = Z \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

$$w_k^c = \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

The above expression is identical to Grad-CAM equation and thus Grad-CAM is a generalization of CAM

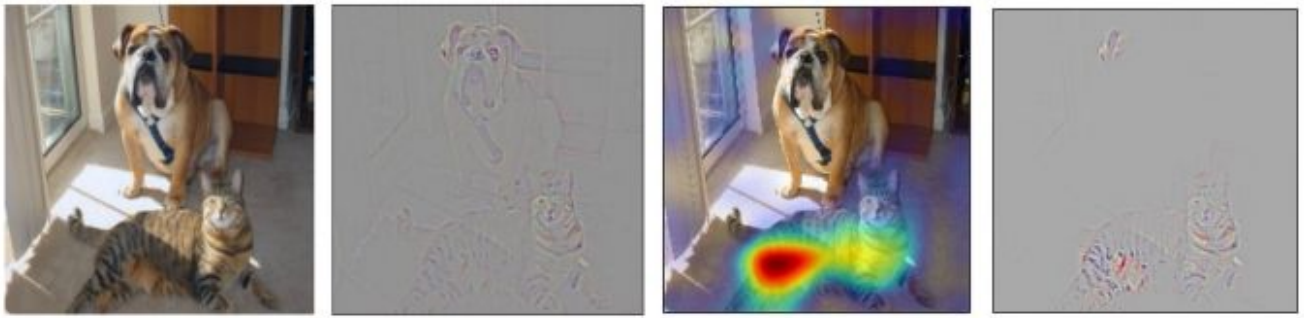
Source [1]

Guided Grad-CAM

Grad-CAM is class discriminative and localizes the model prediction but it lacks the ability to visualize the fine grained details.

Fuse Guided Backprop and Grad-CAM vis element wise multiplication to create Guided Grad-CAM. High resolution and class discriminative.

It highlights the stripes of cat to predict as ‘tiger cat’



(a) Original Image (b) Guided Backprop ‘Cat’ (c) Grad-CAM ‘Cat’ (d)Guided Grad-CAM ‘Cat’

Source [1]

Counterfactual explanation

Using a light modification we can obtain an explanation for the model to change its prediction.

This is achieved a by negating the gradients as shown in the equation.

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

Fig. 3: Counterfactual Explanations with Grad-CAM

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

Source [1]

Weakly supervised localization

Given an image, we first obtain class prediction and then generate Grad-CAM.

Binarize the Grad-CAM using 15% of max intensity as threshold.

Draw a bounding box for the single largest connected segment.

Note, there is no training with annotated bounding box.

		Classification		Localization	
		Top-1	Top-5	Top-1	Top-5
VGG-16	Backprop [51]	30.38	10.89	61.12	51.46
	c-MWP [58]	30.38	10.89	70.92	63.04
	Grad-CAM (ours)	30.38	10.89	**56.51**	46.41
	CAM [59]	33.40	12.20	57.20	**45.14**
AlexNet	c-MWP [58]	44.2	20.8	92.6	89.2
	Grad-CAM (ours)	44.2	20.8	68.3	56.6
GoogleNet	Grad-CAM (ours)	31.9	11.3	60.09	49.34
	CAM [59]	31.9	11.3	60.09	49.34

Table 1: Classification and localization error % on ILSVRC-15 val (lower is better) for VGG-16, AlexNet and GoogleNet. We see that Grad-CAM achieves superior localization errors without compromising on classification performance.

Imagenet localization challenge

Source [1]

Weakly supervised segmentation

Grad-CAM can be used as weak localization seed for image segmentation task.

Intersection of Union Score:

Using CAM as seed : 44.6

Using Grad-CAM: 49.6

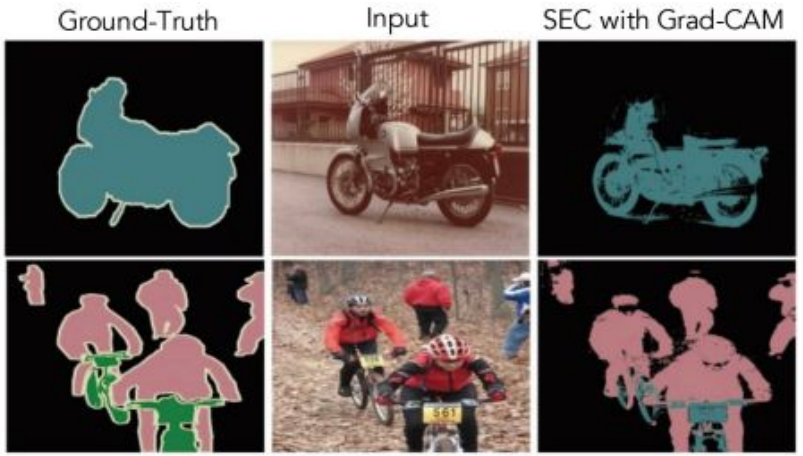


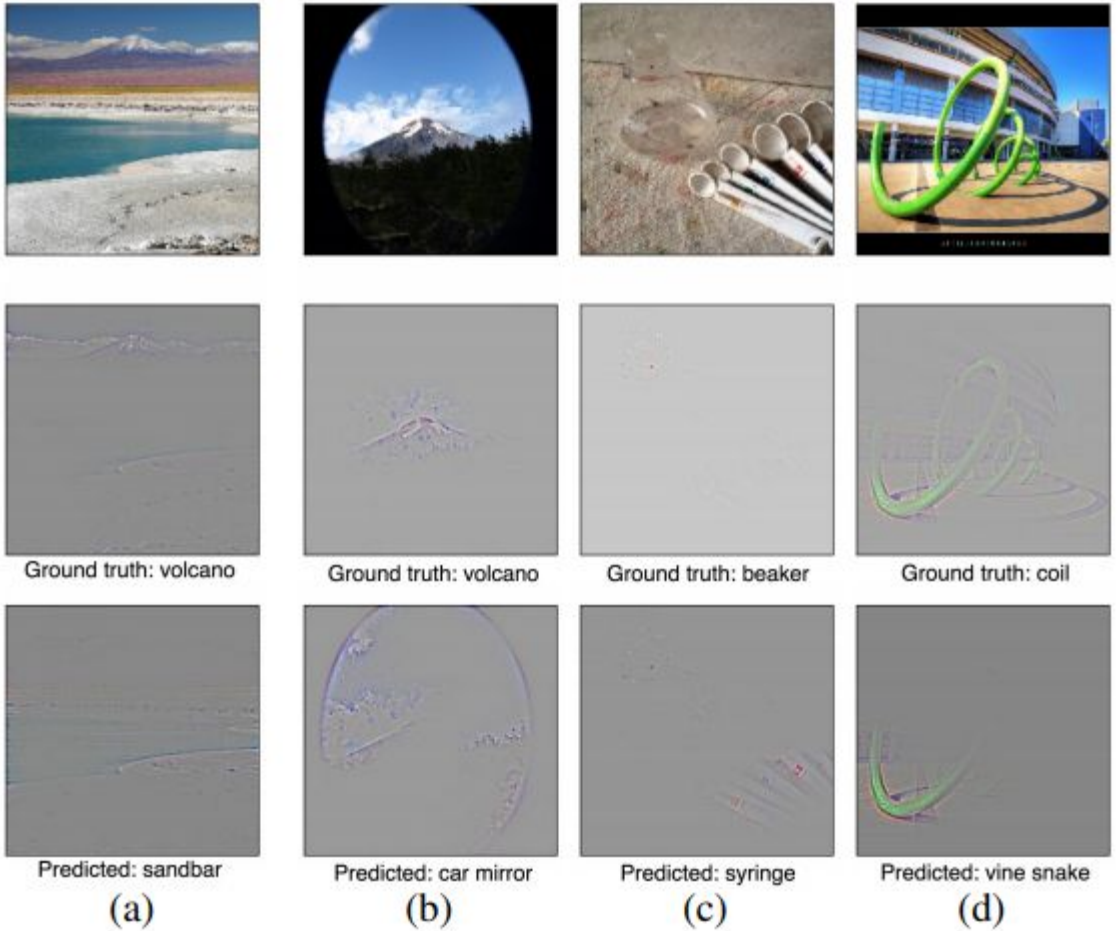
Fig. 4: PASCAL VOC 2012 Segmentation results with Grad-CAM as seed for SEC [32].

PASCAL VOC 2012 Segmentation task

Source [1]

CNNs with Grad-CAM

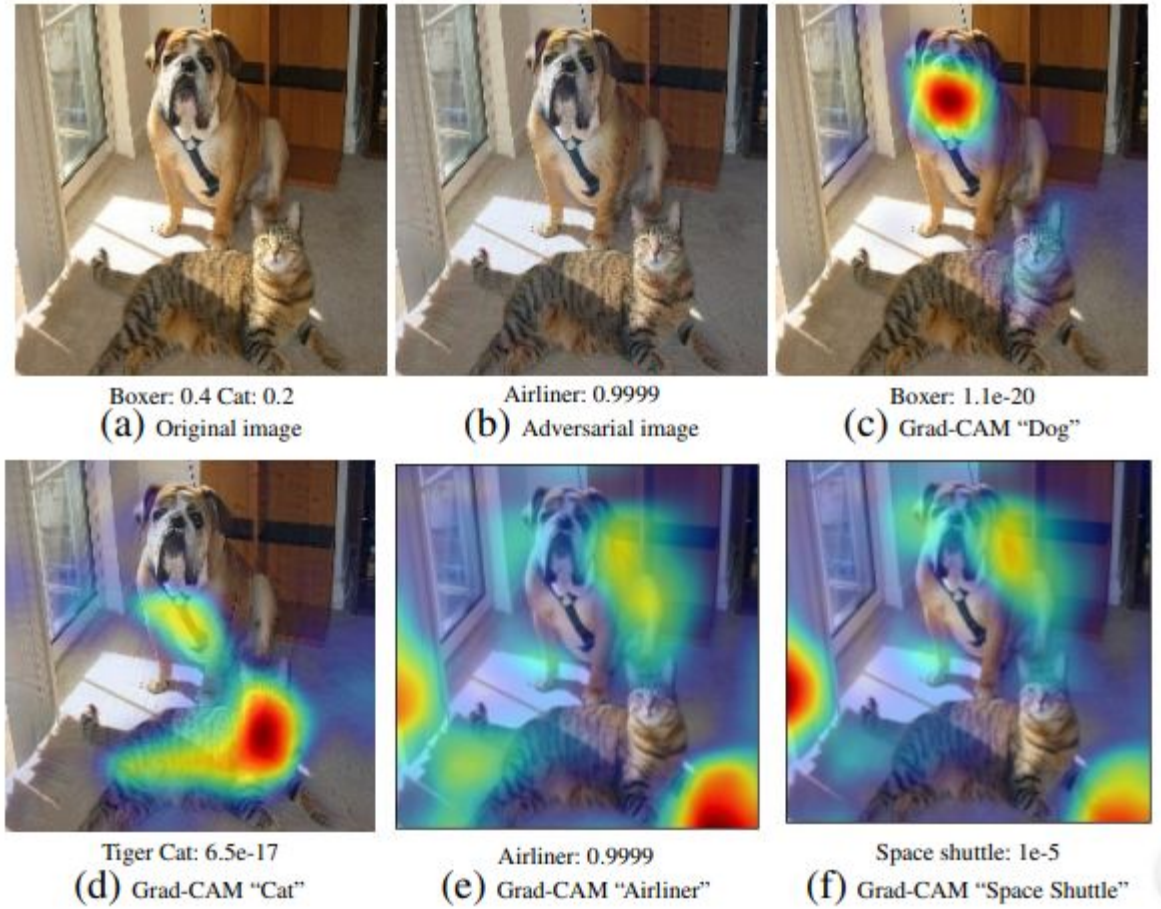
1. **Analyzing the failure modes**
2. Effect of adversarial noise
3. Identifying bias in the dataset



Source [1]

CNNs with Grad-CAM

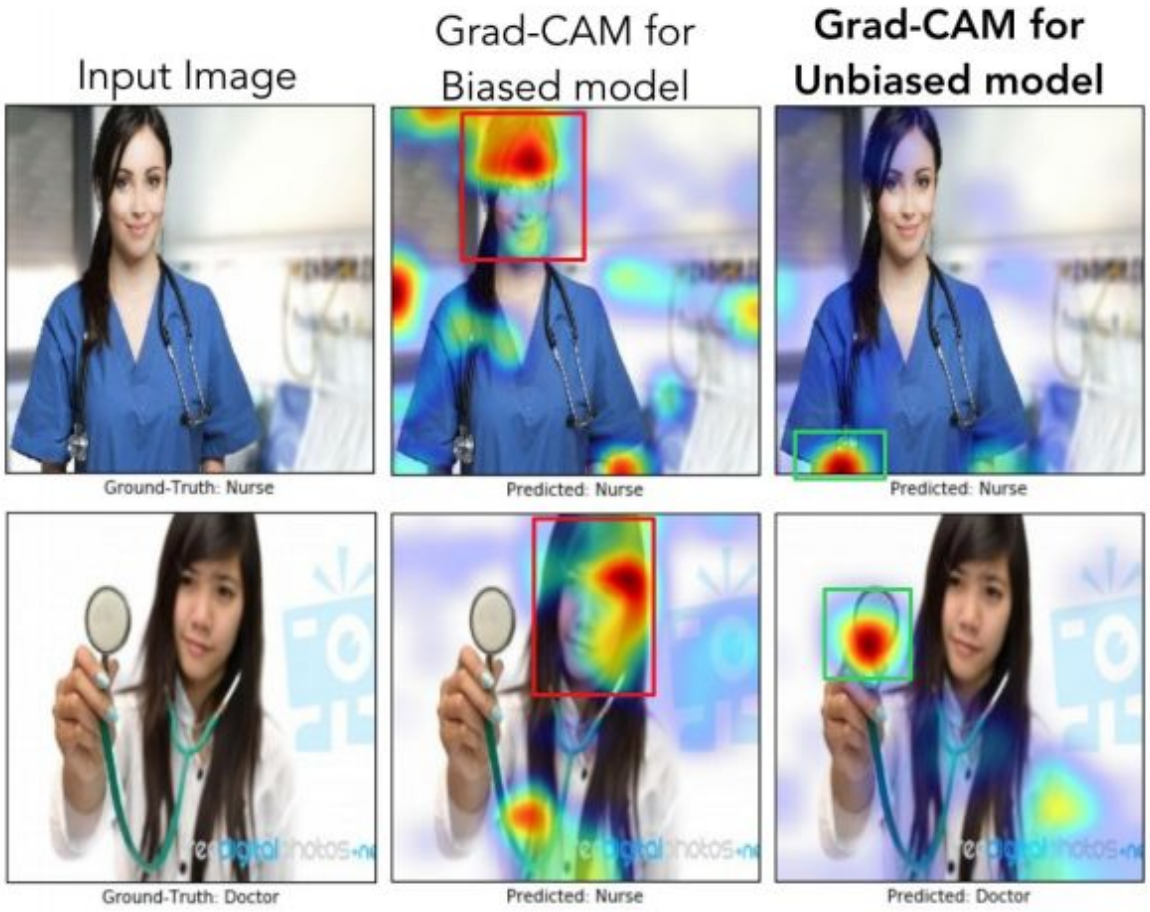
1. Analyzing the failure modes
2. **Effect of adversarial noise**
3. Identifying bias in the dataset



Source [1]

CNNs with Grad-CAM

1. Analyzing the failure modes
2. Effect of adversarial noise
3. **Identifying bias in the dataset**

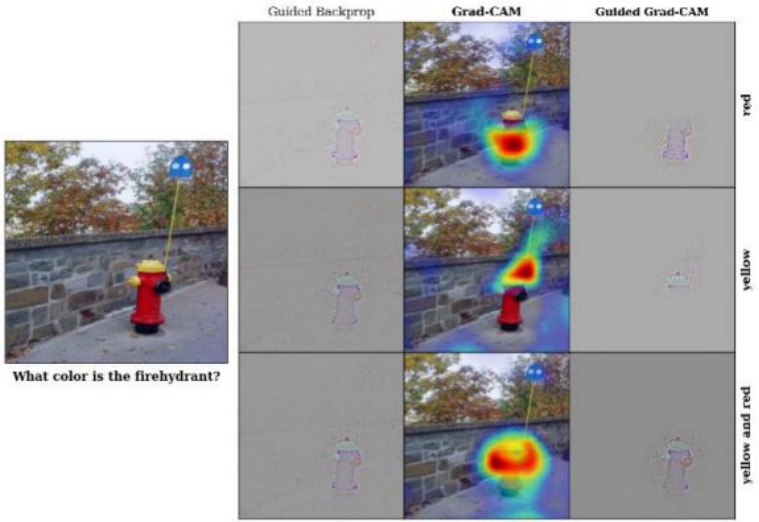


Source [1]

Other applications

1. Image captioning
2. Visual Question Answer

(a) Image captioning explanations



(a) Visualizing VQA model from [38]

Source [1]

PyTorch Hooks

A hook is basically a function that is executed when either forward or backward of *torch.Autograd.Function* (*grad_fn*) is called.

PyTorch provides two hooks:

1. Forward Hook - executed during forward pass
2. Backward Hook - executed during backward pass

PyTorch Hooks

Forward Hook:

```
class Hook():
    def __init__(self, m):
        self.hook = m.register_forward_hook(self.hook_func)
    def hook_func(self, m, i, o): self.stored = o.detach().clone()
    def __enter__(self, *args): return self
    def __exit__(self, *args): self.hook.remove()
```

Backward Hook:

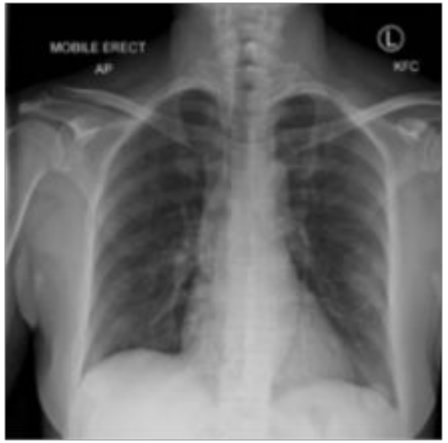
```
class HookBwd():
    def __init__(self, m):
        self.hook = m.register_backward_hook(self.hook_func)
    def hook_func(self, m, gi, go): self.stored = go[0].detach().clone()
    def __enter__(self, *args): return self
    def __exit__(self, *args): self.hook.remove()
```

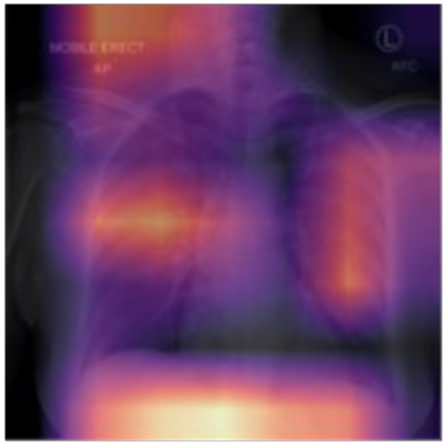
Chest X-Ray Model Interpretation

Grad-CAM can be used in medical image diagnosis to explain radiologists or doctors the model prediction.

Here an x-ray of COVID-19 patient is shown and the Grad-CAM heat map explain why the model predicted COVID-19.

The model looks in the lungs but also look in the text on top left.





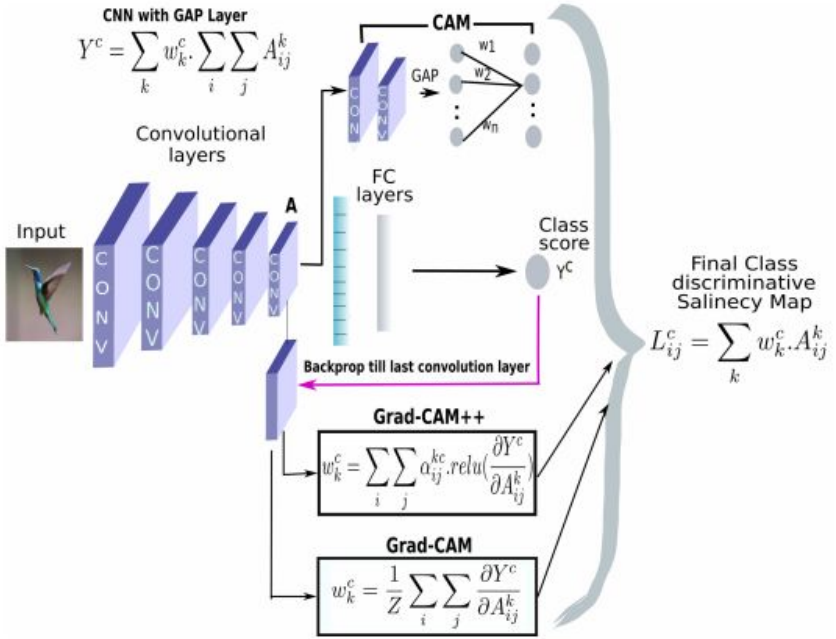
Grad-CAM++

Generalization of Grad-CAM to solve two issues.

Grad-CAM++ claims to have better interpretation when the image contains multiple occurence of same class.

Localization covers the entire object.

Global average is replaced by weighted average of gradients



$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \quad \Rightarrow \quad w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

Source [6]

References

1. https://arxiv.org/pdf/1610.02391.pdf
2. https://arxiv.org/pdf/1512.04150.pdf
3. https://github.com/ramprs/grad-cam
4. https://medium.com/@mohamedchetoui/grad-cam-gradient-weighted-class-activation-mapping-ffd72742243a
5. https://github.com/vickyliin/gradcam_plus_plus-pytorch
6. https://arxiv.org/pdf/1710.11063.pdf
7. https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/

Code Demo

https://colab.research.google.com/drive/1ybtEWZctzHwC84OcU8kiUlB4BH2VD1dC?usp=sharing