Deep Learning for Computer Vision

# Explaining CNNs: Class Attribution Map Methods

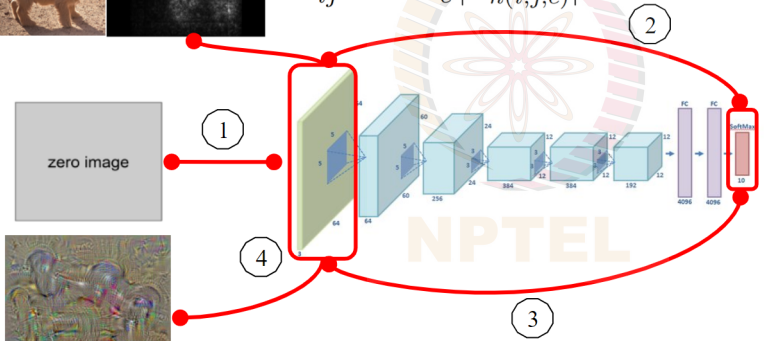
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

# Going Beyond Optimization-to-Image Methods

Visualize the data gradient. (Note that the data has 3 channels)

$$M_{ij} = \max_c |w_{h(i,j,c)}|$$




Find images that maximize some class score.

1) Feed in zeros.

2) Forward the image through the network.

3) Set the gradient of the score vector to be [0,0,..,1,..0] and backprop to image.

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

[Score for class c (before softmax)]

4) Do a small image update.

5) Go back to 2.

# Going Beyond Optimization-to-Image Methods

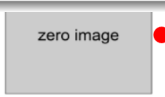
Visualize the data gradient. (Note that the data has 3 channels)

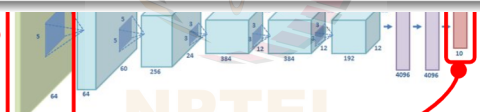
$$M_{ij} = \max_c |w_{h(i,j,c)}|$$

1) Feed in zeros.

**Question**

Can we know what a network was looking at, while predicting a class?



2) score vector to be [0,0,..,1,..0] and backprop to image.

$$\arg\max_I \boxed{S_c(I)} - \lambda \|I\|_2^2$$

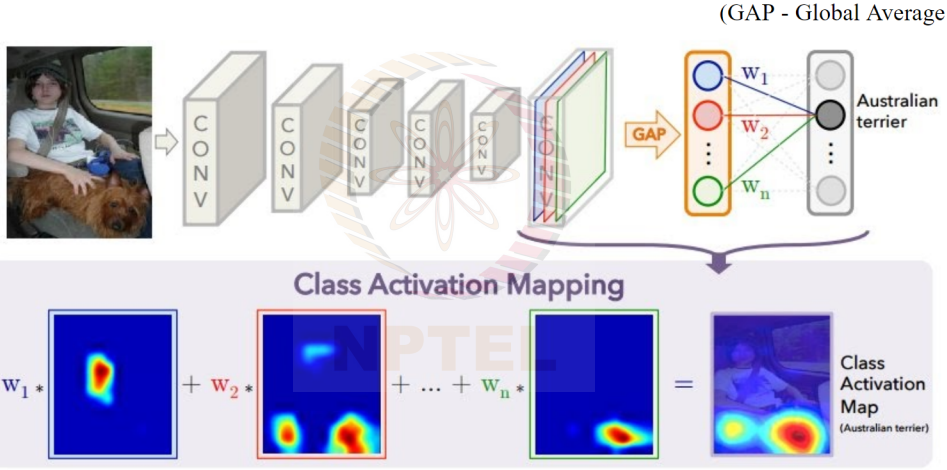
[Score for class c (before softmax)]

3) Do a small image update.

4) Go back to 2.

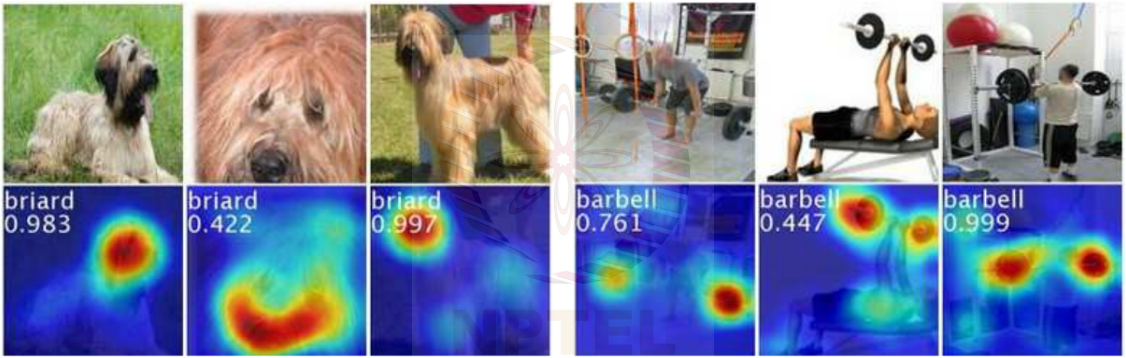
Find images that maximize some class score.

# Class Activation Maps (CAM)[1]



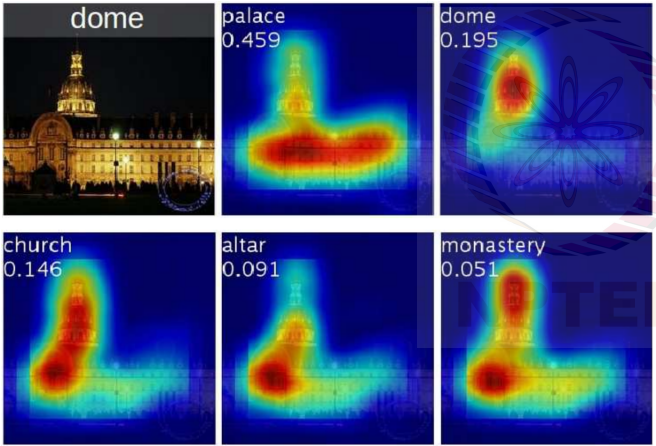

[1] Zhou et al, Learning Deep Features for Discriminative Localization, CVPR 2016

# CAM: Examples




Discriminative image regions used for classification of "Briard" and "Barbells" classes. In the first set, the model is using the dog's face to make the decision and in the second set, it is using the weight plates.

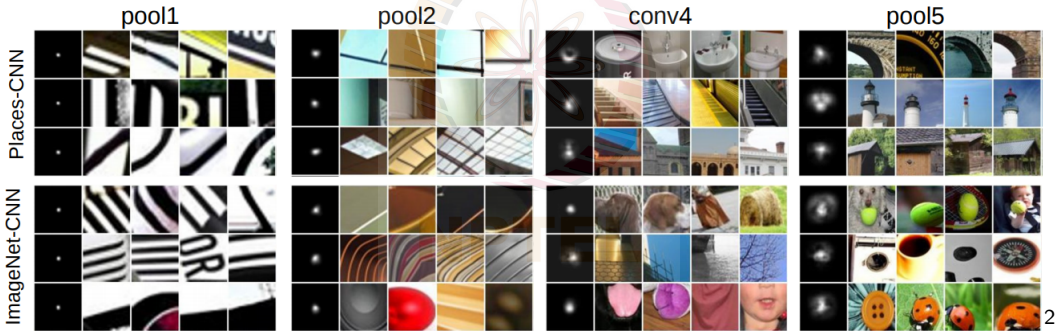
# CAM: Examples




Top 5 predicted classes and their corresponding CAMs. Notice how the same activation maps produce different CAMs based on weights connecting features to individual classes

# CAM: Intuition

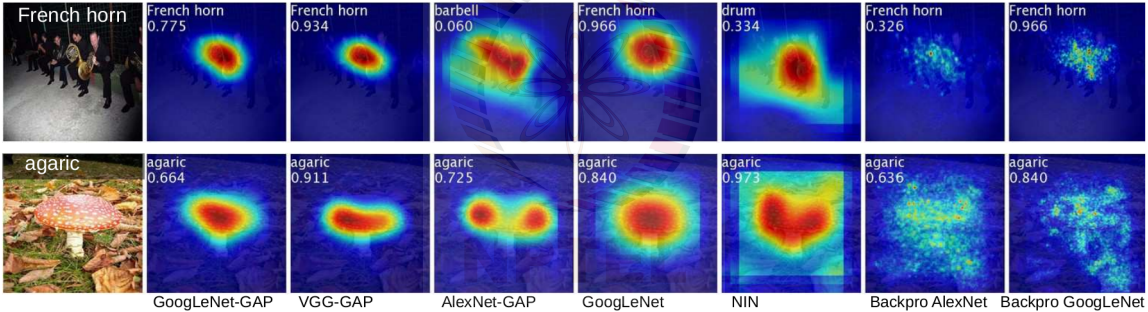
Convolutional units behave as object localizers even without supervision over objects' location; this capability is lost if fully connected layers are used for classification




Receptive fields of convolutional units and their maximally activating image patch examples

[2] Zhou et al, Object Detectors emerge in Deep Scene CNNs, ICLR 2015

# CAM: Comparison



# CAM: Pros and Cons

# CAM: Pros and Cons

**Advantages**

- Is class discriminative (can localize objects without positional supervision).
- Doesn't require a backward pass unlike guided backprop or deconvolution

# CAM: Pros and Cons

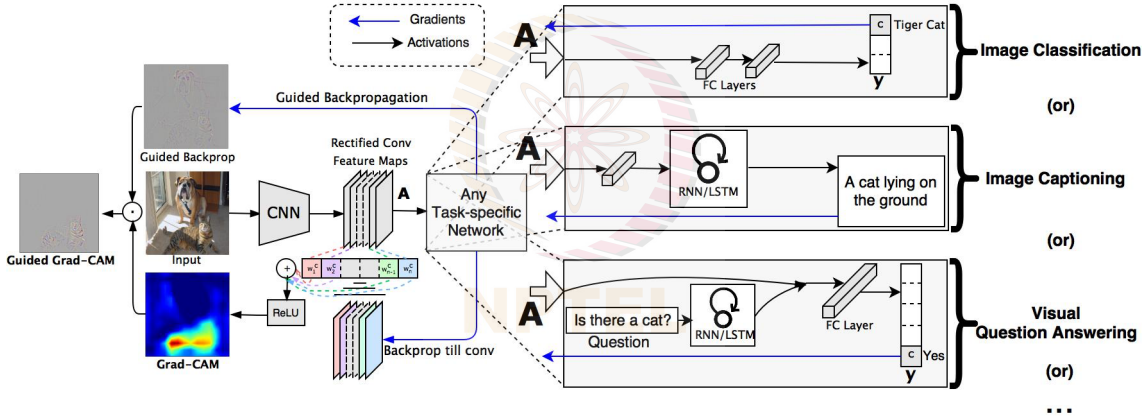
**Advantages**

- Is class discriminative (can localize objects without positional supervision).
- Doesn't require a backward pass unlike guided backprop or deconvolution

**Disadvantages**

- Constraint on architecture is restrictive; may not be useful to explain complex tasks like image captioning or visual question answering (VQA)
- Model may trade off accuracy for interpretability
- Need for retraining to explain trained models

# Gradient-weighted CAM (Grad-CAM)[3]




[3]Selvaraju et al, Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, ICCV 2017

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

  where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$ ; then, $Y^c = \sum_k w_k^c \cdot F^k$;

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$ ; then, $Y^c = \sum_k w_k^c \cdot F^k$; we then have:

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

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

$$\frac{\partial Y^c}{\partial F^k} = w_c^k = \frac{\partial Y^c}{\partial A_{ij}^k} \cdot Z$$

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$ ; then, $Y^c = \sum_k w_k^c \cdot F^k$; we then have:

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

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

$$\frac{\partial Y^c}{\partial F^k} = w_c^k = \frac{\partial Y^c}{\partial A_{ij}^k} \cdot Z$$

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

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$ ; then, $Y^c = \sum_k w_k^c \cdot F^k$; we then have:

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

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$ ; then, $Y^c = \sum_k w_k^c \cdot F^k$; we then have:

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

$$\frac{\partial Y^c}{\partial F^k} = w_c^k = \frac{\partial Y^c}{\partial A_{ij}^k} \cdot Z$$

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

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

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$ ; then, $Y^c = \sum_k w_k^c \cdot F^k$; we then have:

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

$$\frac{\partial Y^c}{\partial F^k} = w_c^k = \frac{\partial Y^c}{\partial A_{ij}^k} \cdot Z$$

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

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

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

# Grad-CAM: Generalization of CAM

- From CAM, we have:

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

where $A_{ij}^k$ is the pixel at $(i, j)$ location of $k$th feature map

- Let $F^k = \frac{1}{Z}\sum_i \sum_j A_{ij}^k$ ; then, $Y^c = \sum_k w_k^c \cdot F^k$; we then have:

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

$$\frac{\partial Y^c}{\partial F^k} = w_c^k = \frac{\partial Y^c}{\partial A_{ij}^k} \cdot Z$$

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

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

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

Class-feature weights are gradients themselves.
No retraining required!

# Grad-CAM: Methodology

- Uses gradients flowing from output class into activation maps of last convolutional layer as neuron importance weights ($w_k^c$).

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

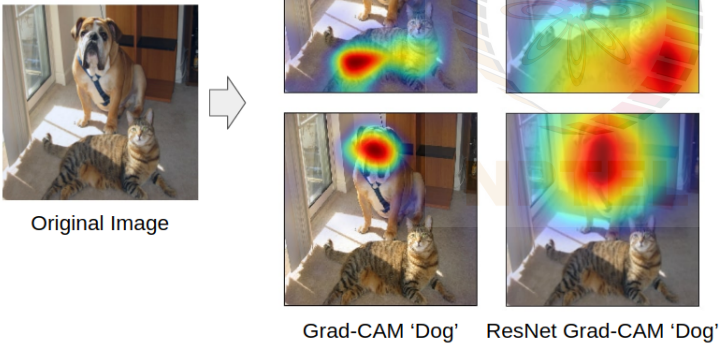
where $w_k^c$ = weight of $k^{th}$ activation map w.r.t class c

$A^k = k^{th}$ activation map

- Siimilar to CAM, localization map $L_{Grad-CAM}^c$ is given by:

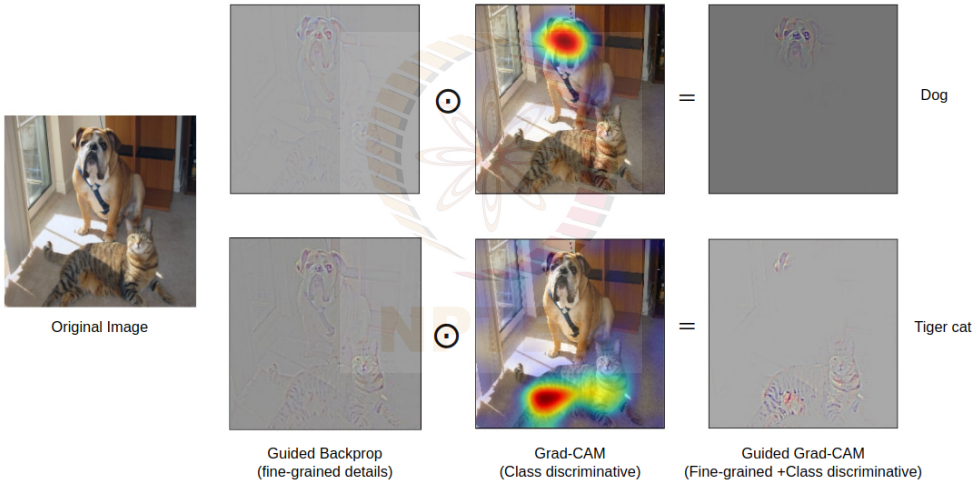
$$L_{Grad-CAM}^c = ReLU\Big(\sum_k w_k^c A^k\Big)$$

# Grad-CAM: Example




- Grad-CAM maps are **class-discriminative**
- However, it is **unclear** from this heat-map why the network predicts this particular instance as **'tiger cat'**
- Can we do something about this?

# Guided Grad-CAM

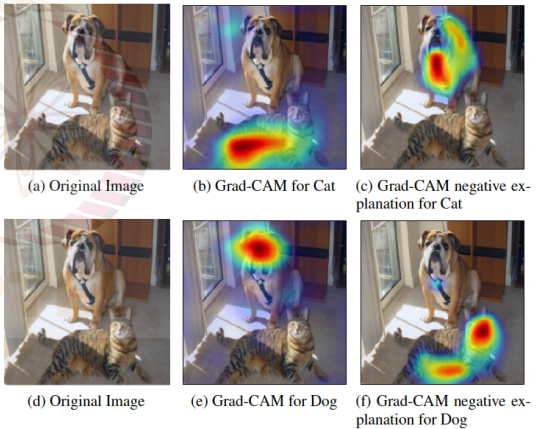


# Grad-CAM: Counterfactual Explanations

- Negating the value of gradients used for calculation of importance weights ($w_k^c$) causes localization maps to show image patches that adversarially affect classification output

$$w_k^c = \frac{1}{Z} \sum_i \sum_j -\frac{\partial Y^c}{\partial A_{ij}^k}$$

- Removing/suppressing features occurring in such patches can improve model confidence

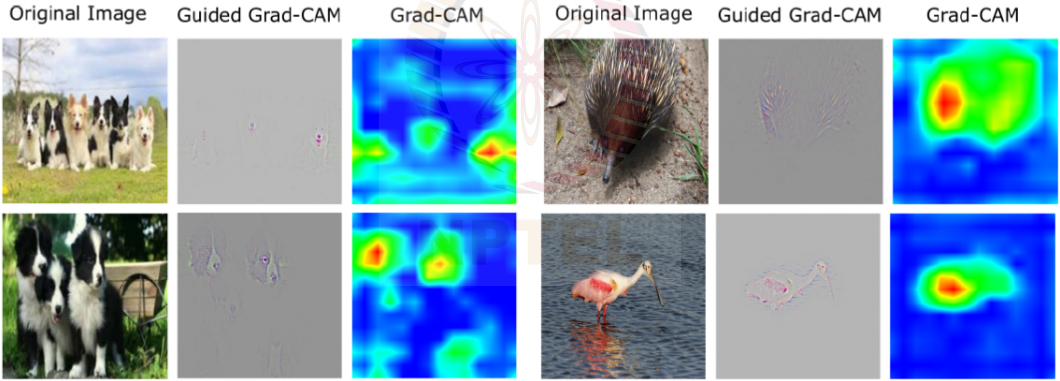


(a) Original Image (b) Grad-CAM for Cat (c) Grad-CAM negative explanation for Cat

(d) Original Image (e) Grad-CAM for Dog (f) Grad-CAM negative explanation for Dog

# Grad-CAM: Limitations[4]

- Inability to identify multiple instances of objects
- Unsatisfactory localization performance, especially under occlusion




[4] Chattopadhay et al, Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks, WACV 2018

# Grad-CAM++: Motivation

- Grad-CAM considers all pixel gradients equally when computing importance weights of activation maps

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

# Grad-CAM++: Motivation

- Grad-CAM considers all pixel gradients equally when computing importance weights of activation maps

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

- This can suppress activation maps with comparatively lesser spatial footprint

# Grad-CAM++: Motivation

- Grad-CAM considers all pixel gradients equally when computing importance weights of activation maps

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

- This can suppress activation maps with comparatively lesser spatial footprint
- Since instances of objects in an image tend to have different shapes and orientations, some of them can fade away

# Grad-CAM++: Motivation

- Grad-CAM considers all pixel gradients equally when computing importance weights of activation maps

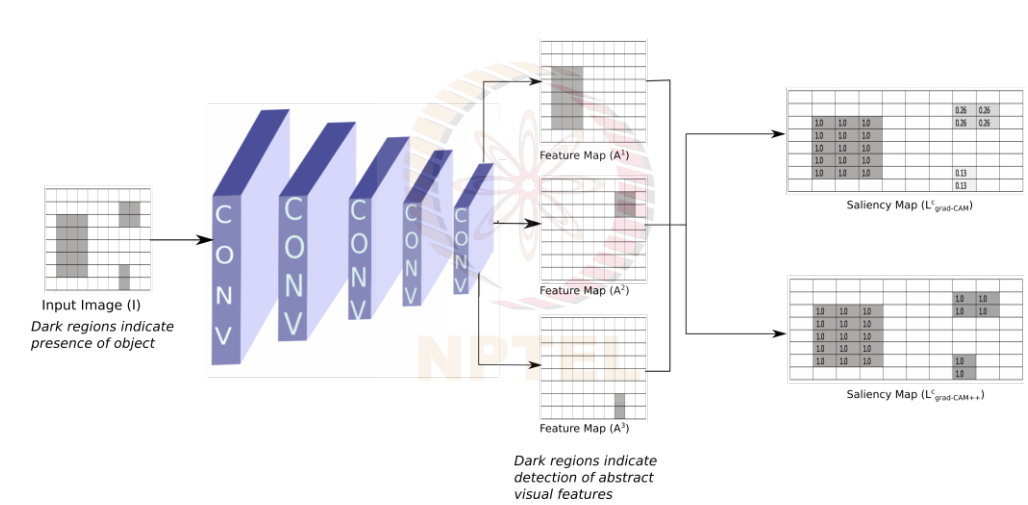
$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

- This can suppress activation maps with comparatively lesser spatial footprint
- Since instances of objects in an image tend to have different shapes and orientations, some of them can fade away
- This can be corrected by using weighted average of pixel-wise gradients

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \underbrace{ReLU\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right)}_{\text{Focus on positive gradients}}$$

where $\alpha$ is the pixel-wise weight. How to find $\alpha$?

# Grad-CAM++: Intuition



# Grad-CAM++: Methodology

- For a particular class c and activation map k, the pixel-wise weight $\alpha^{kc}$ at pixel position $(i, j)$ can be calculated as:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}$$

NOTE: Both a,b and i,j are iterators on the same activation map. They are only used to avoid confusion. How? Homework![5]

[5] Section 3.1 to 3.4, Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

# Grad-CAM++: Methodology

- For a particular class c and activation map k, the pixel-wise weight $\alpha^{kc}$ at pixel position $(i, j)$ can be calculated as:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}$$

  NOTE: Both a,b and i,j are iterators on the same activation map. They are only used to avoid confusion. How? Homework![5]

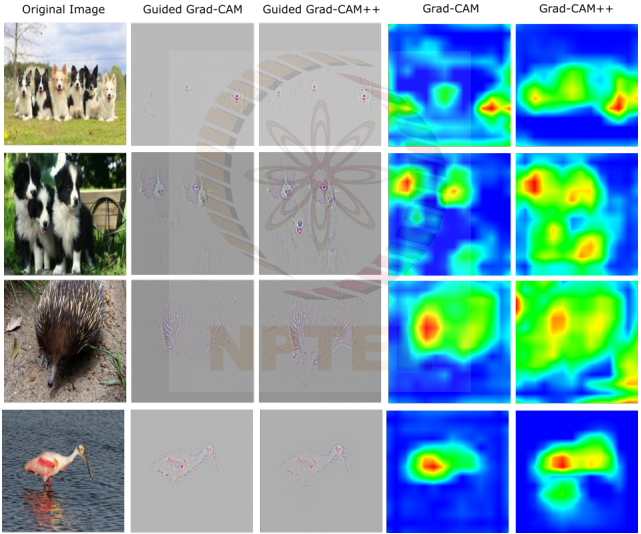
- Final localization map $L_{Grad-CAM++}$ (similar to that of GradCAM):

$$L_{Grad-CAM++} = ReLU\Big(\sum_k w_k^c A^k\Big)$$

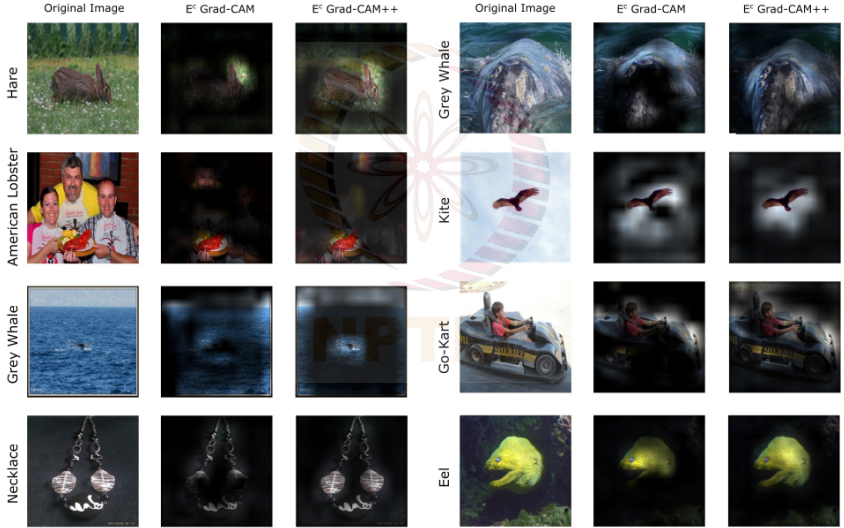
$$\text{where } w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} ReLU\Big(\frac{\partial Y^c}{\partial A_{ij}^k}\Big)$$

[5] Section 3.1 to 3.4, Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

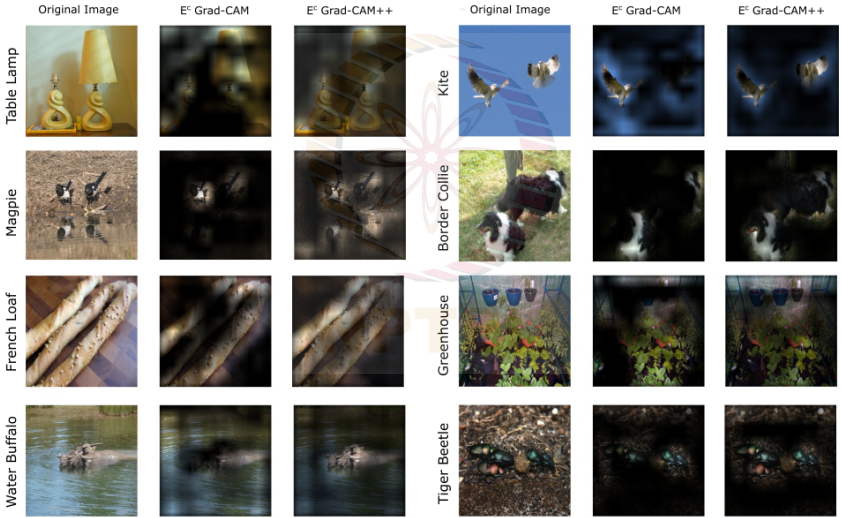
# Grad-CAM++: Example



# Grad-CAM++: Examples for Class Localization



# Grad-CAM++: Examples for Multiple Occurrences



# Homework

## Readings

- Zhou et al, Learning Deep Features for Discriminative Localization, CVPR 2016
- Selvaraju et al, Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, ICCV 2017
- Chattopadhyay et al, Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks, WACV 2018

# References


Min Lin, Qiang Chen, and Shuicheng Yan. "Network in network". In: *arXiv preprint arXiv:1312.4400* (2013).

Bolei Zhou et al. "Object detectors emerge in deep scene cnns". In: *arXiv preprint arXiv:1412.6856* (2014).

Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

Bolei Zhou et al. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

Aditya Chattopadhay et al. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 839–847.