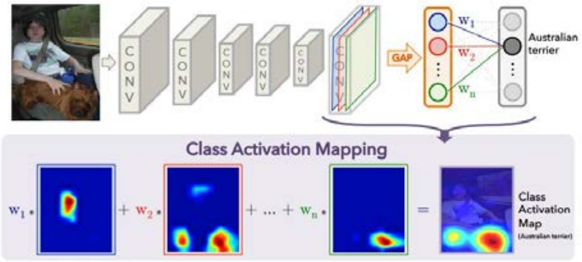
Class Activation Map series
2019/01/14

Data and Visual Analytics Lab

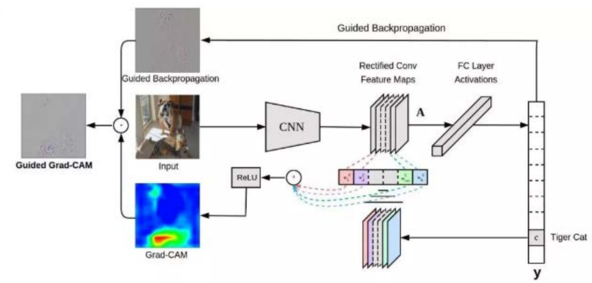
김강열

CAM Series

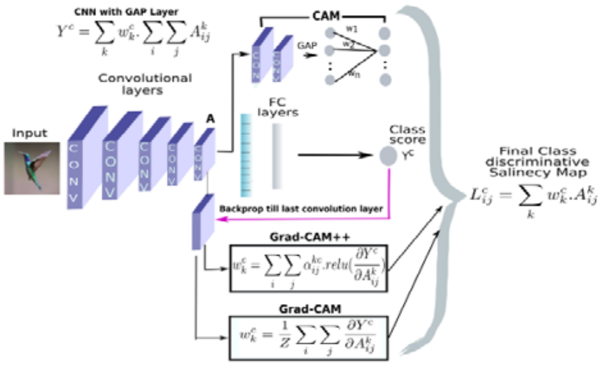
- Learning Deep Features for Discriminative Localization(i.e. CAM)



- Grad-CAM: Gradient-weighted Class Activation Mapping

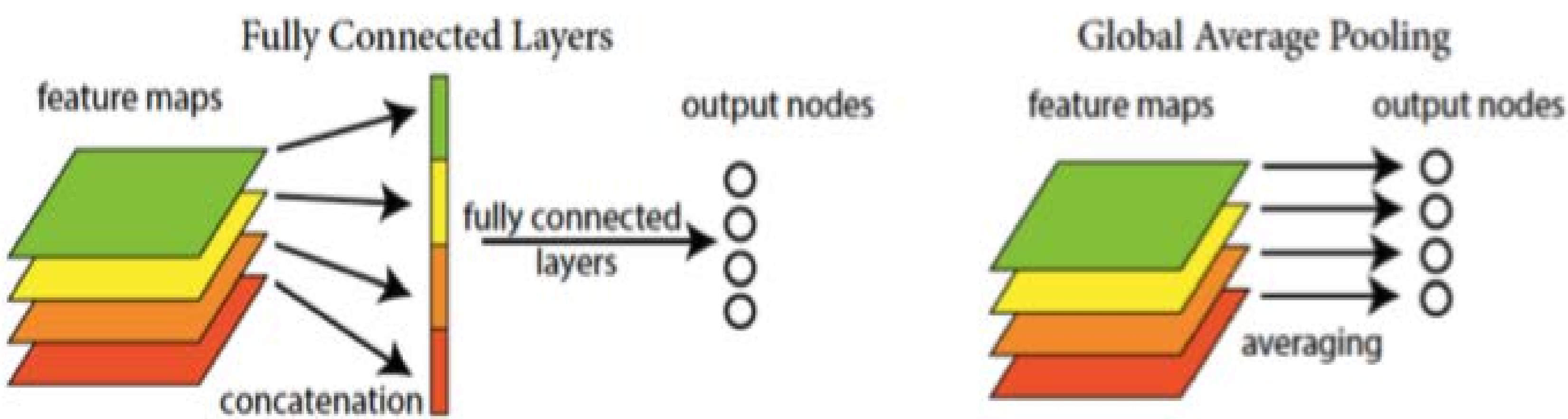


- Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks



Global Average Pooling

- Network In Network에서 처음 제안됨 ← Fully connected layers are prone to overfitting
  - No parameter to optimize(parameter▼) → Overfitting 방지
  - Average pooling은 spatial한 정보를 합하는 방식이기 때문에 입력 이미지의 spatial 변환에 robust
  - feature map들과 category 사이에 직관적인 관계 해석 가능



Class Activation Mapping

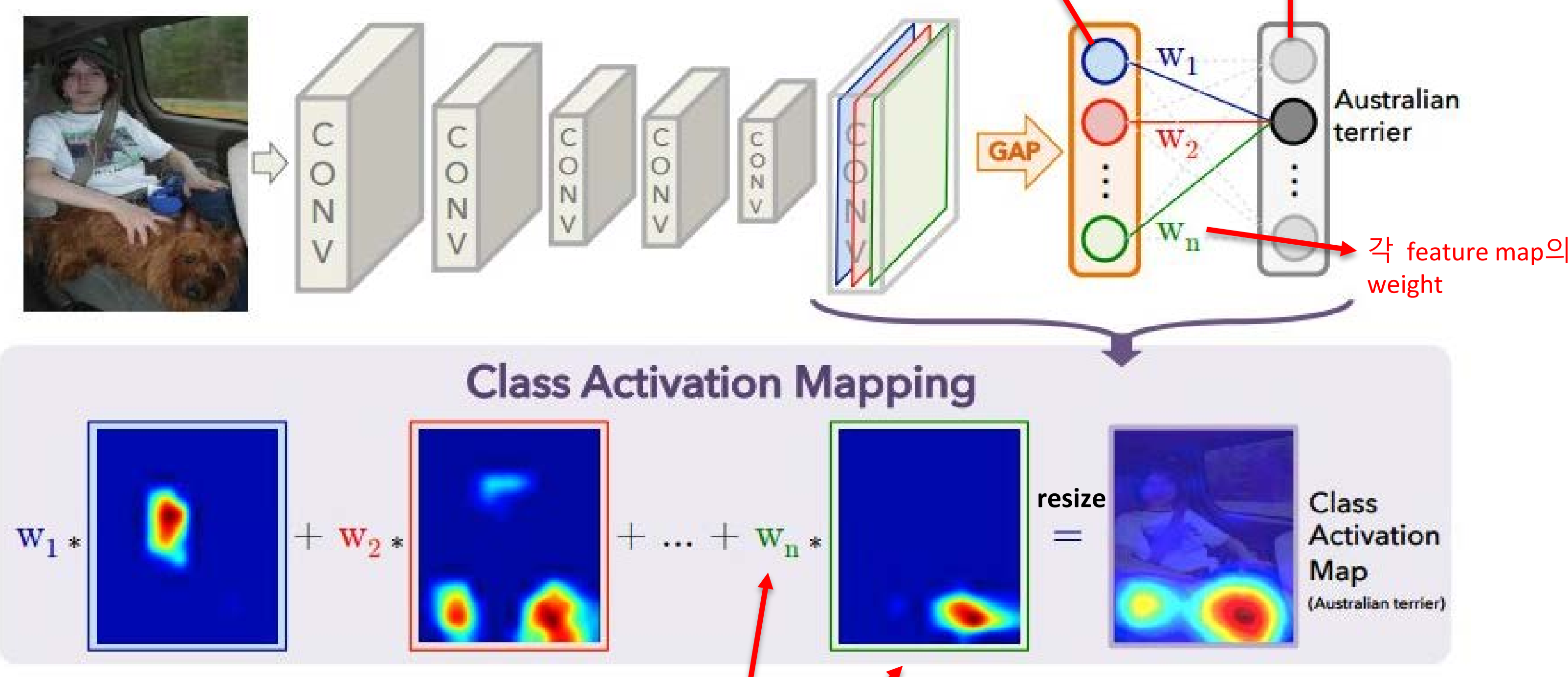
Class score

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y)$$

$$= \sum_{x,y} \boxed{\sum_k w_k^c f_k(x,y)}$$

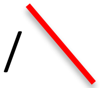
CAM

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



CAM $$M_c(x,y) = \sum_k w_k^c f_k(x,y).$$

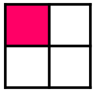
Class Activation Mapping Intuition

4x4 input / 2x2 2-filters with stride=2 / $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ /  ,  2-classes

$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{convolution op}} \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

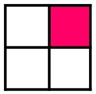
$w_1^1 \text{x } 2 + w_2^1 \text{x } 0$

$w_1^1 > w_2^1$; pattern detector로써 filter ~ 해당 class에 맞는 filter에 대응해서 가중치가 증가

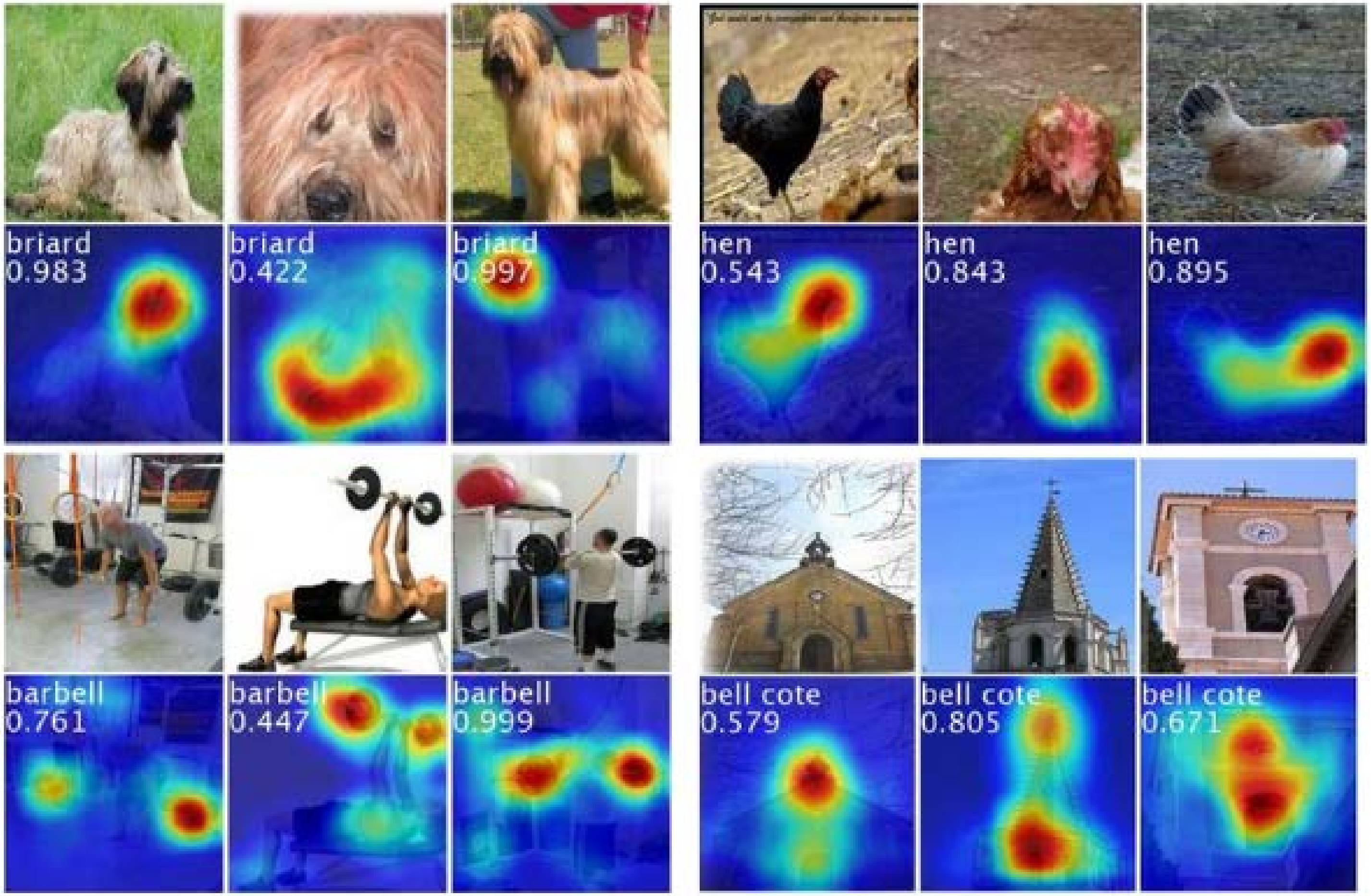
$w_1^1 \text{x} \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + w_2^1 \text{x} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \rightarrow$ 

$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{convolution op}} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$

$w_1^2 \text{x } 0 + w_2^2 \text{x } 2$

$w_1^2 \text{x} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + w_2^2 \text{x} \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \rightarrow$ 

Result



briard
0.983
briard
0.422
briard
0.997
hen
0.543
hen
0.843
hen
0.895
barbell
0.761
barbell
0.447
barbell
0.999
bell cote
0.579
bell cote
0.805
bell cote
0.671

Limitations => Grad-CAM

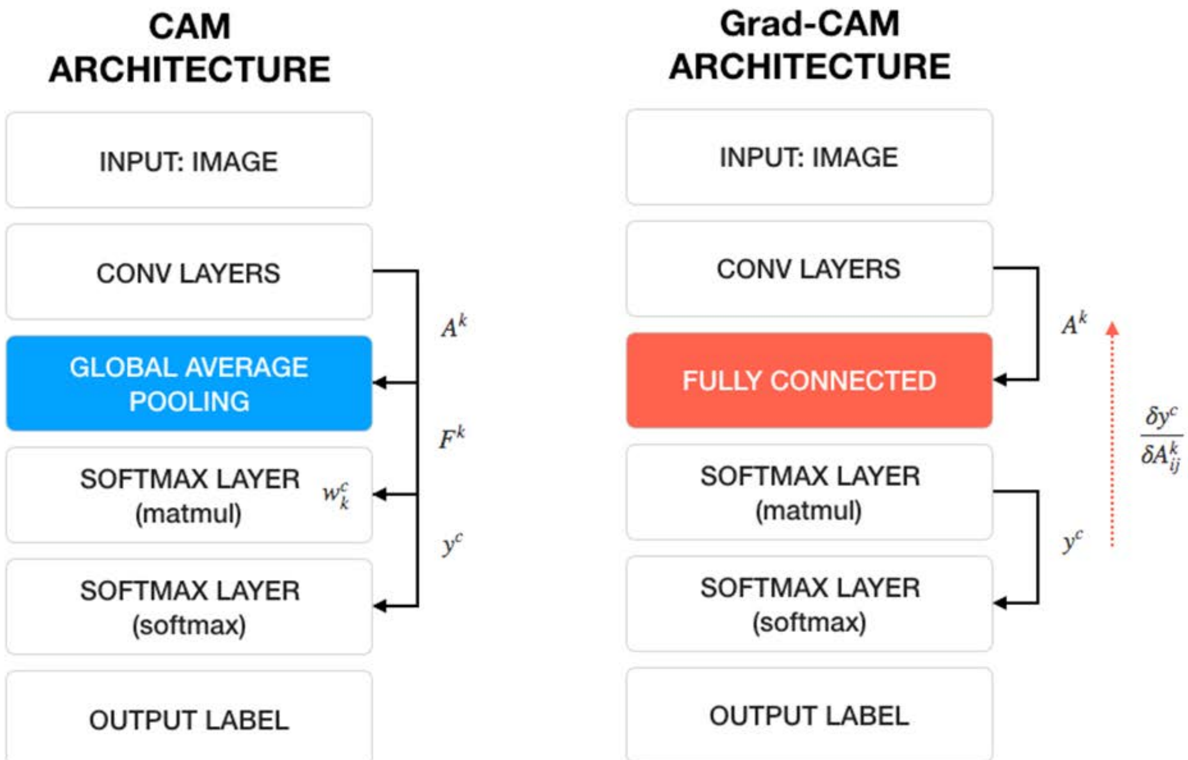
- CAM trades off model complexity and performance for more transparency into the working of the model => Grad-CAM does not need to alter model architecture

- CAM must perform GAP preceding softmax layer(feature maps -> GAP -> softmax)
- => Grad-CAM could use intermediate feature maps to compute the gradient value

- CAM is applied on the limited area such as classification => Grad-CAM broadened the areas to captioning, VQA or reinforce learning

- Grad-CAM is a generalization of CAM

CAM vs. Grad–CAM

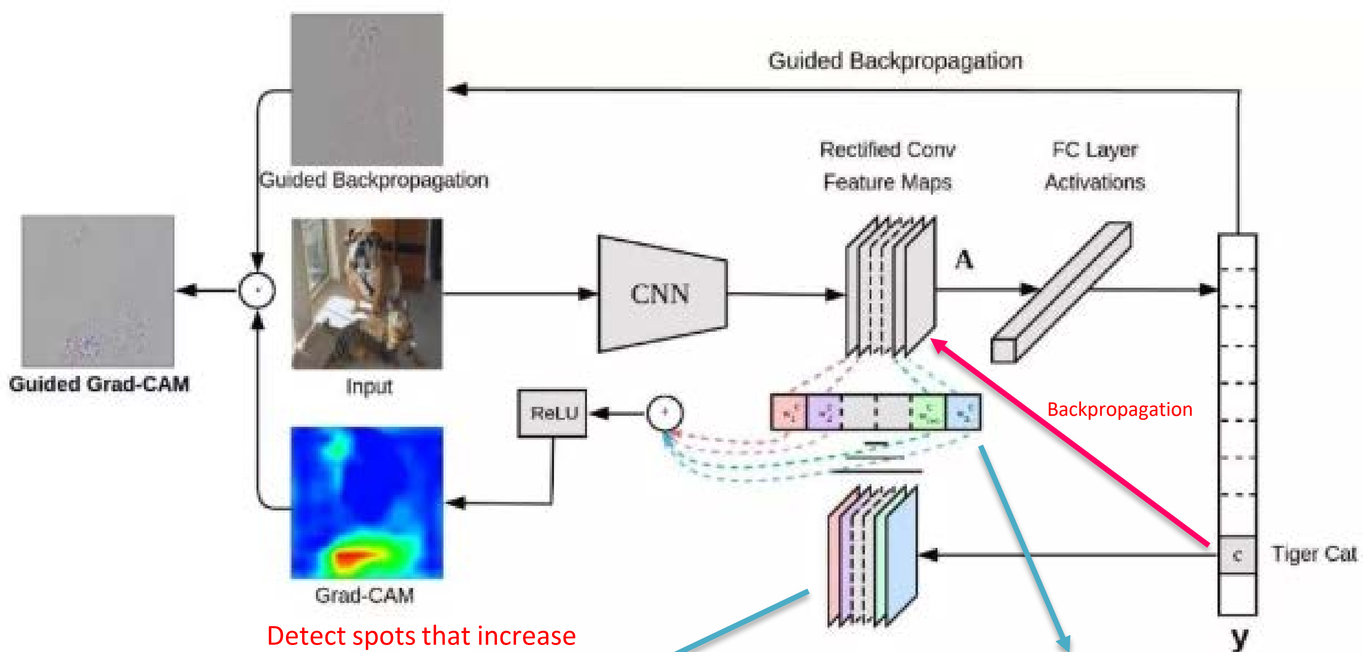


$$L_{CAM}^c = \sum_k w_k^c A^k$$

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

$$a_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\delta y^c}{\delta A_{ij}^k}$$

Overall Structure



Detect spots that increase pixel value

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

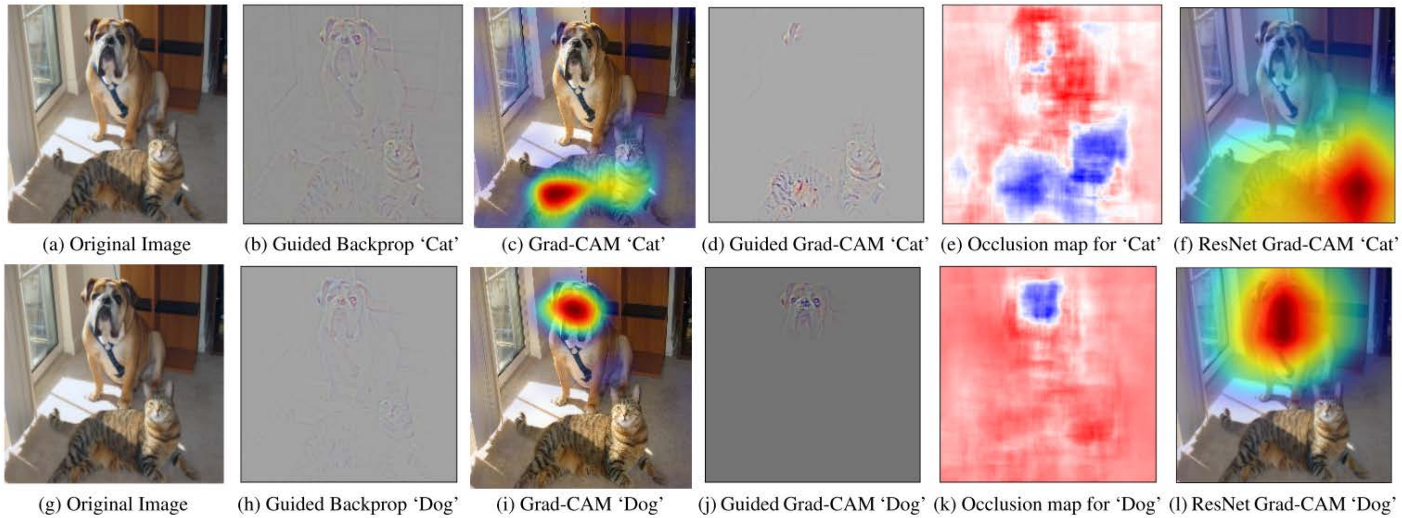
Weight of feature map

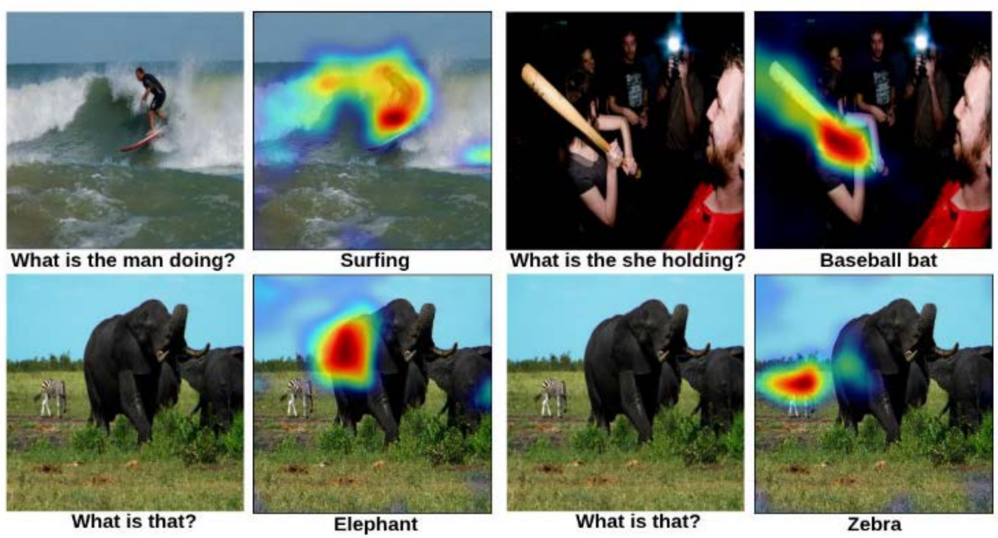
$$\alpha^c_k = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backprop}}$$

Implications:

- Given filters, As indicate where the patterns appear and the intensity of it. Thus, the maps generated by fitted filter would make higher value of it.
- It's obvious then the value of alpha(i.e. weight of feature map) would amplify with the fitted filter
- $ReLU$ after linear combination: detect only regions that pixel should increase

Result


(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'


What is the man doing?
Surfing
What is the she holding?
Baseball bat
What is that?
Elephant
What is that?
Zebra

Limitations => Grad-CAM++

- Grad-CAM’s results are not frequently precise or cannot cover all area of object => Grad-CAM++ generates better result with well-covered heatmap
- Grad-CAM++ formulated generation process strictly
- Grad-CAM introduced pixel-wise weighting of the gradients of the output w.r.t. the final convolutional feature map of the CNN
- Propose new metric for evaluating the faithfulness of the proposed explanations to the underlying model.
- Propose a training methodology to involve newly generated explanation image in the relationship between teacher and student(Knowledge distillation field)

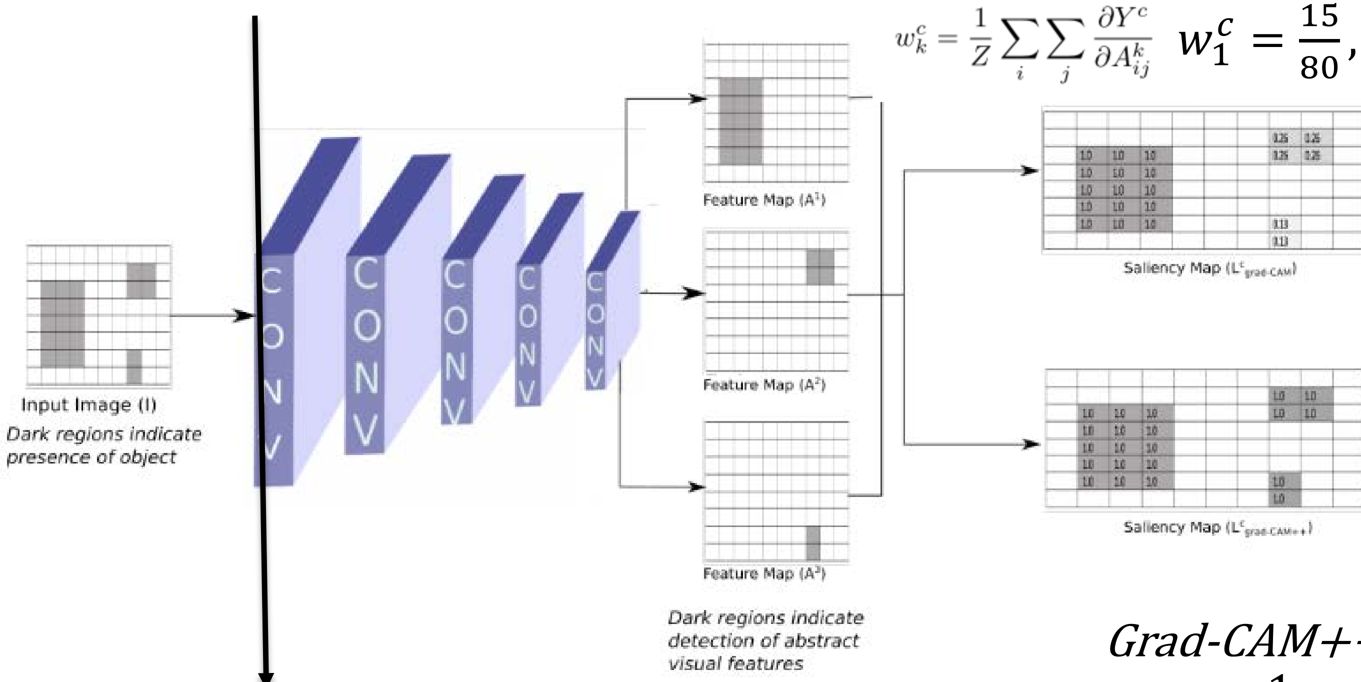
Why Grad-CAM++? – Multi objects perception

$$\frac{\partial y^c}{\partial A_{ij}^k} = 1 \quad if \quad A_{ij}^k = 1$$
$$= 0 \quad if \quad A_{ij}^k = 0$$

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

Grad-CAM weight:

$$w_1^c = \frac{15}{80}, w_2^c = \frac{4}{80}, w_3^c = \frac{2}{80},$$



proportional to object size

equal regardless of object size

$$\alpha_{ij}^{kc} = \frac{1}{\sum_{l,m} \frac{\partial y^c}{\partial A_{lm}^k}} \quad if \quad \frac{\partial y^c}{\partial A_{ij}^k} = 1$$
$$= 0 \quad \text{otherwise}$$

pixel-wise weighted average

$$L_{ij}^c = \sum_k w_k^c . A_{ij}^k$$
$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} . relu\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right)$$

Grad-CAM++ weight:

$$\alpha^{1c} = \frac{1}{15}, \alpha^{2c} = \frac{1}{4}, \alpha^{3c} = \frac{1}{2}$$
$$w_1^c = 1, w_2^c = 1, w_3^c = 1,$$

Why Grad-CAM++? – Coarse Heatmap

Cannot catch **FULL** body of snake

Cannot catch **FULL** body of bird

Why Grad-CAM++? – Coarse Heatmap

Why? Activated

0	0	0
4	0	4
0	0	0

Eyes

0	0	0
0	0	0
0	100	0

Mouth

Grad-CAM

$$L_{ij}^{c} = \sum_{k} w_{k}^{c}.A_{ij}^{k}$$

Normalization term

$$w_k^c = \sum_i \sum_j [\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \underline{\sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}}].relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

$$L_{ij}^{c} = \sum_{k} w_{k}^{c}.A_{ij}^{k}$$

Some Computations for alpha and w

Reformulation

$$Y^c = \sum_k w_k^c . \sum_i \sum_j A_{ij}^k$$

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

$$\Rightarrow \quad Y^c = \sum_k \{\sum_a \sum_b \alpha_{ab}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ab}^k})\}[\sum_i \sum_j A_{ij}^k]$$

Computation Simplicity

$$Y^c = \exp(S^c) \qquad \frac{\partial Y^c}{\partial A_{ij}^k} = \exp(S^c)\frac{\partial S^c}{\partial A_{ij}^k}$$

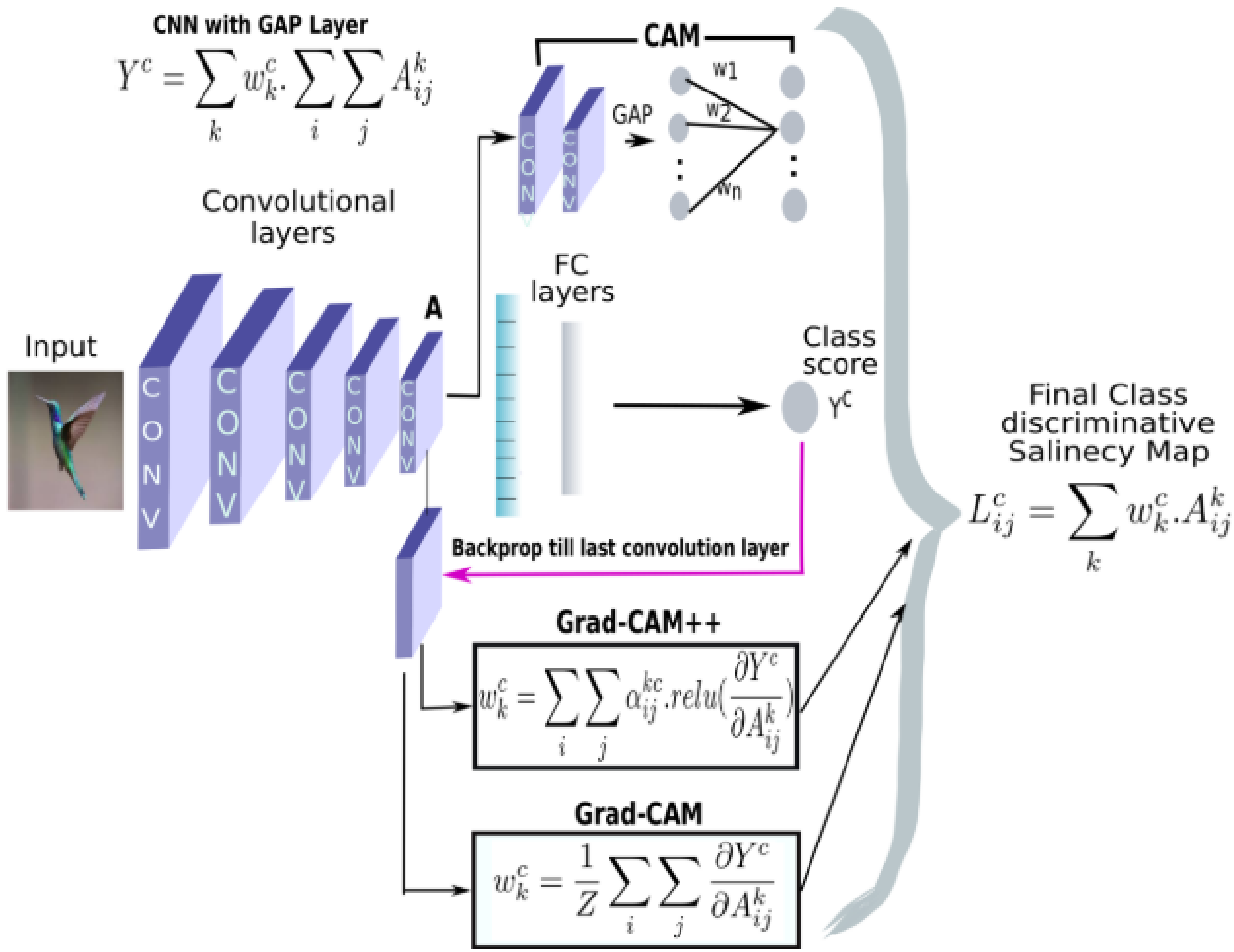
$$\Rightarrow \quad \frac{\partial Y^c}{\partial A_{ij}^k} = \sum_a \sum_b \alpha_{ab}^{kc} . \frac{\partial Y^c}{\partial A_{ab}^k} + \sum_a \sum_b A_{ab}^k \{\alpha_{ij}^{kc} . \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}\}$$

Result

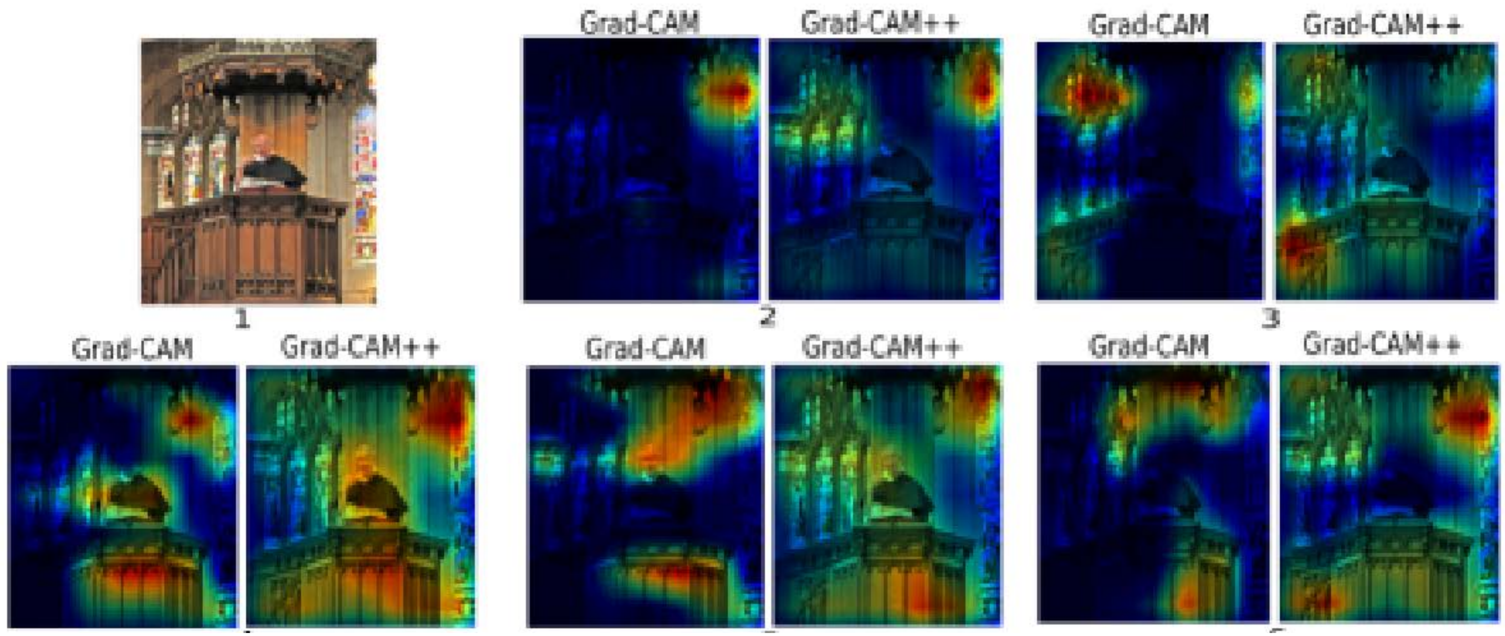
$$\Rightarrow \quad \alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}$$

$$w_k^c = \sum_i \sum_j [\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}].relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

Overall Structure



Result


Grad-CAM
Grad-CAM++
Grad-CAM
Grad-CAM++
1
2
3
Grad-CAM
Grad-CAM++
Grad-CAM
Grad-CAM++
Grad-CAM
Grad-CAM++


A
Original Image
E^c Grad-CAM
E^c Grad-CAM++
Original Image
E^c Grad-CAM
E^c Grad-CAM++
Cottontail Rabit
Partridge
Grey Whale
Monitor