Evaluating Explainable CNN Techniques : CAM,gradCAM,gradCAM++

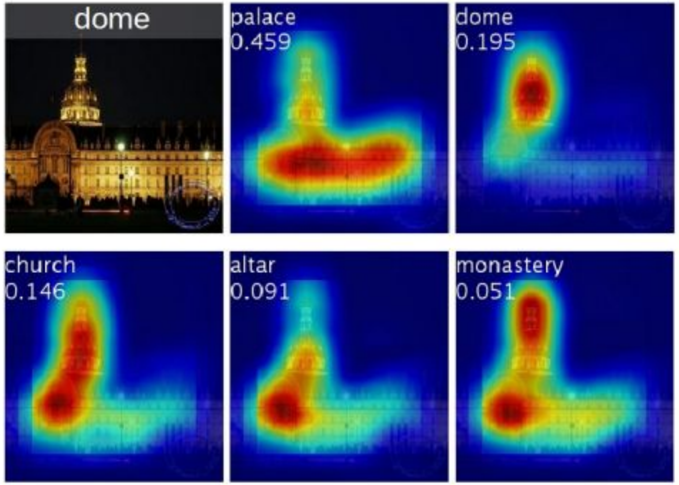
Presented by :
Monalisa Bakshi (21697)
Nikhil Bisen (21698)

Introduction

- Deep learning models are reckoned as “black-box” unlike classical machine learning.
- There is a need for Explainable Techniques for Convolutional Neural Networks which can tell us what our neural network is looking at while giving the output .
- Explainability become very crucial in AI for HealthCare domains , where doctors would be able to diagnose the disease more accurately if they have the model interpretability available .
- In 2016 we had the first explainability methods coming out - Class Activation Maps for Networks with Global Average Pooling and then soon after GradCAM which solve some issues of CAM .
- In 2017 , the modified version of gradCAM , gradCAM++ comes out and after that there have been series of development in Explainable CNN with different CAM’s model coming out
- We are going to see the working of CAM , gradCAM and gradCAM++ today , understand the experimental setup that we did to evaluate this models .

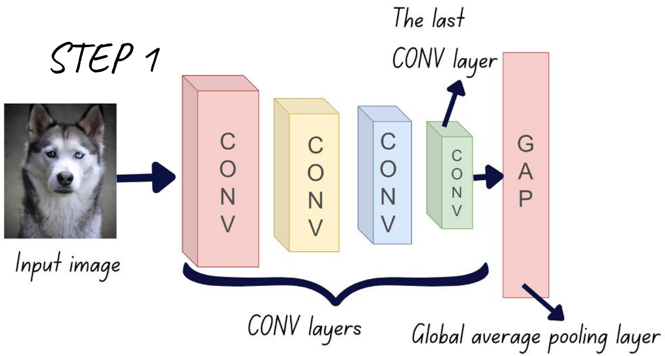
Class Activation Mappings(CAM)

- Convolutionals units behave as object localizers even without supervision over objects location ; this capability Is lost if fully connected layers are used for classification
- Side Figure : Examples of the CAMs generated from the top 5 predicted categories for the given image
- the highlighted regions vary across predicted classes



Steps to Generate Activation Maps :

STEP 1



num Feature Maps = num Filters in CONV4 layer = n

Feature map #l

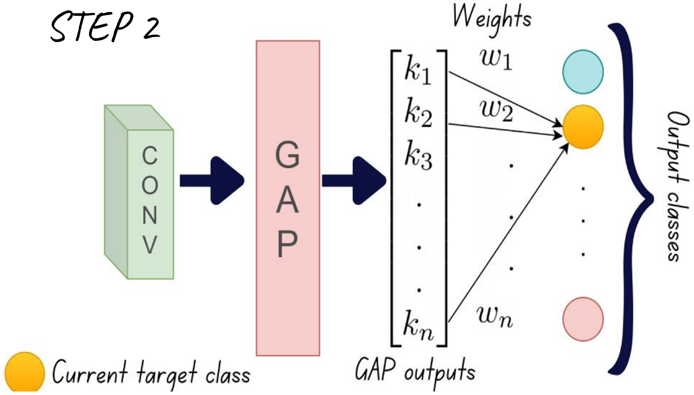
a1	a2	a3
a4	a5	a6
a7	a8	a9

$$\rightarrow \quad k1 = \frac{a1 + a2 + \ldots + a9}{9}$$

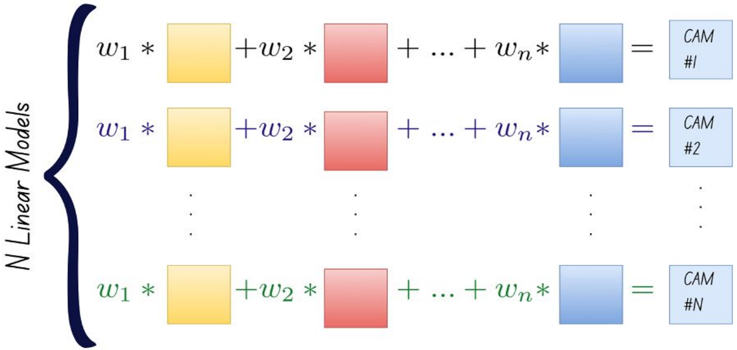
Global Average Pooling

Repeat for feature map #2 through #n.

STEP 2



STEP 3

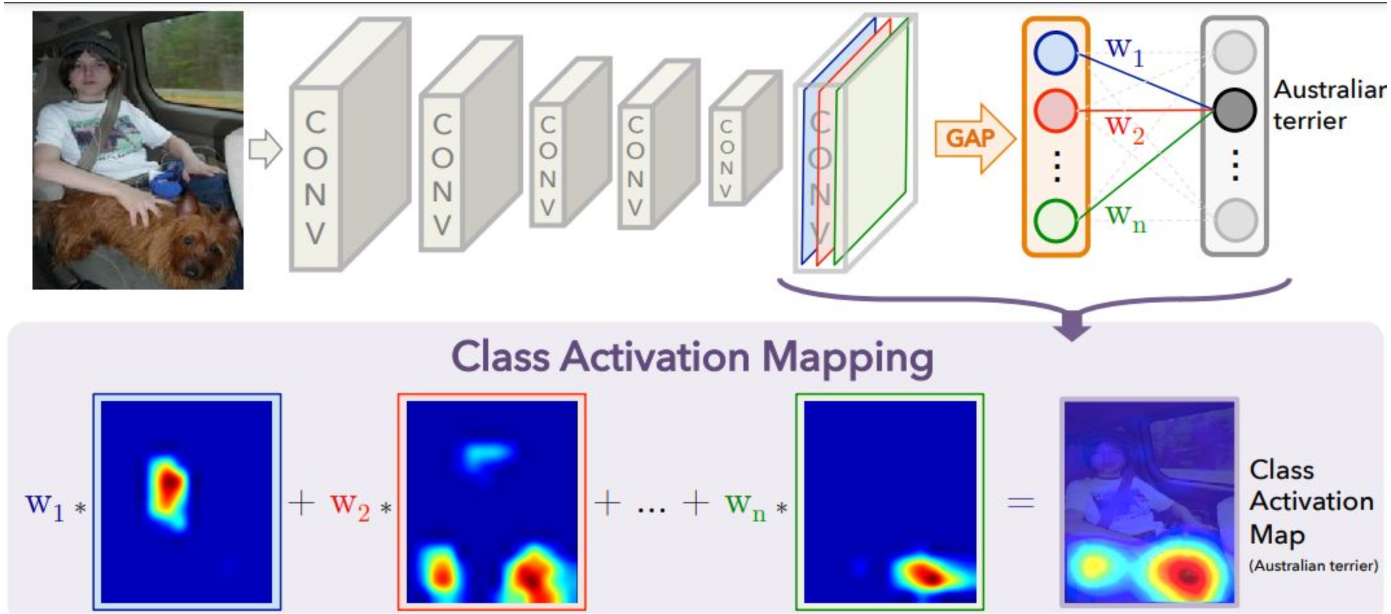

N Linear Models
$w_1 * \quad + w_2 * \quad + \ldots + w_n * \quad =$ CAM #l
$w_1 * \quad + w_2 * \quad + \ldots + w_n * \quad =$ CAM #2
$w_1 * \quad + w_2 * \quad + \ldots + w_n * \quad =$ CAM #N

$$y^c = \sum_k w_k{}^c \frac{1}{Z} \sum_i \sum_j A_{ij}{}^k \quad (1)$$

$A_{ij}^k :$ *pixel at location* (i, j) *in the* $k-th$ *feature map*

$Z :$ *total number of pixels in the feature map*

$w_k{}^c :$ *weight of the* $k-th$ *feature map for class* c

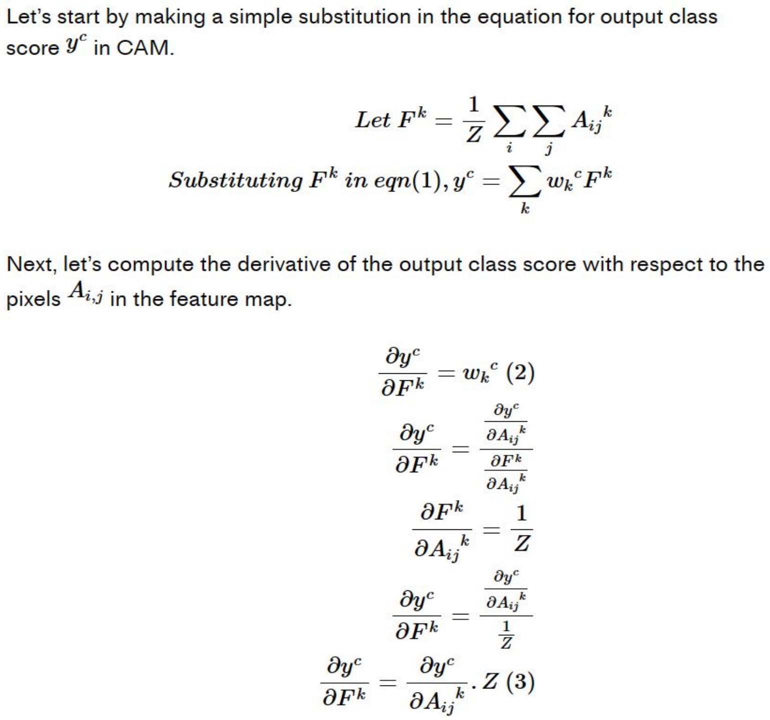


Limitations of CAM :

- Using class activation maps involves the overhead of learning N linear models to learn the weights w_1,w_2,w_3…..wn for each of the N classes
- The introduction of the global average pooling (GAP) layer after the last convolutional layer imposes a *restriction* on the ConvNet architecture.
- CAM can only be used in Classification Task and can not be used in more complex tasks like Visual question answering(VQA) and Image Captioning .

GradCAM : generalization of CAM

As mentioned, the key limitation of CAM is the overhead of learning the weights for linear mapping.
Gradient-weighted class activation map (Grad-CAM) is a generalization to CAM that overcomes this limitation.



Let's start by making a simple substitution in the equation for output class score y^c in CAM.

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

$$Substituting\ F^k\ in\ eqn(1), y^c = \sum_k {w_k}^c F^k$$

Next, let's compute the derivative of the output class score with respect to the pixels $A_{i,j}$ in the feature map.

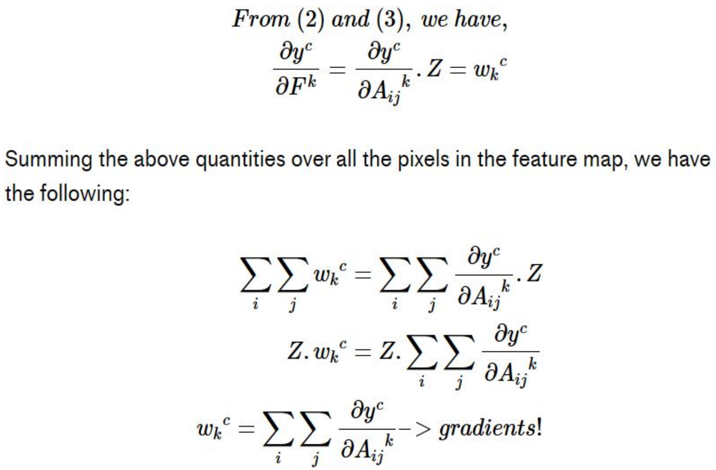
$$\frac{\partial y^c}{\partial F^k} = {w_k}^c\ (2)$$

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

$$\frac{\partial F^k}{\partial {A_{ij}}^k} = \frac{1}{Z}$$

$$\frac{\partial y^c}{\partial F^k} = \frac{\frac{\partial y^c}{\partial {A_{ij}}^k}}{\frac{1}{Z}}$$

$$\frac{\partial y^c}{\partial F^k} = \frac{\partial y^c}{\partial {A_{ij}}^k}.Z\ (3)$$



$$From\ (2)\ and\ (3),\ we\ have,$$

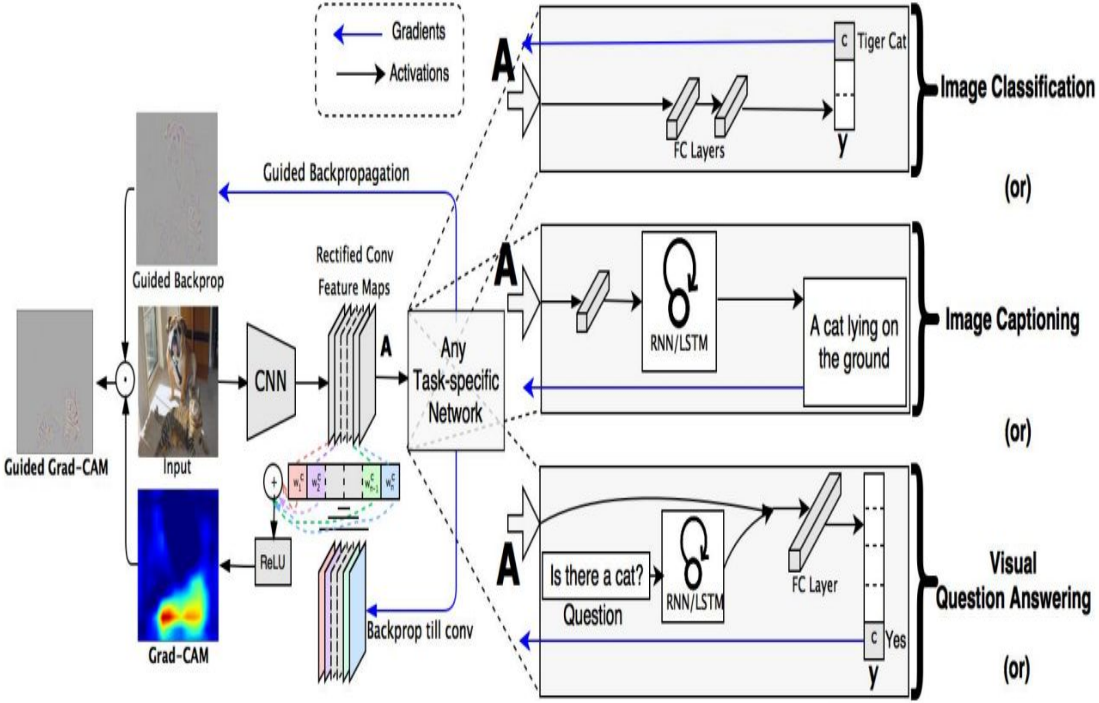
$$\frac{\partial y^c}{\partial F^k} = \frac{\partial y^c}{\partial {A_{ij}}^k}.Z = {w_k}^c$$

Summing the above quantities over all the pixels in the feature map, we have the following:

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

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

$${w_k}^c = \sum_i\sum_j \frac{\partial y^c}{\partial {A_{ij}}^k} -> gradients!$$

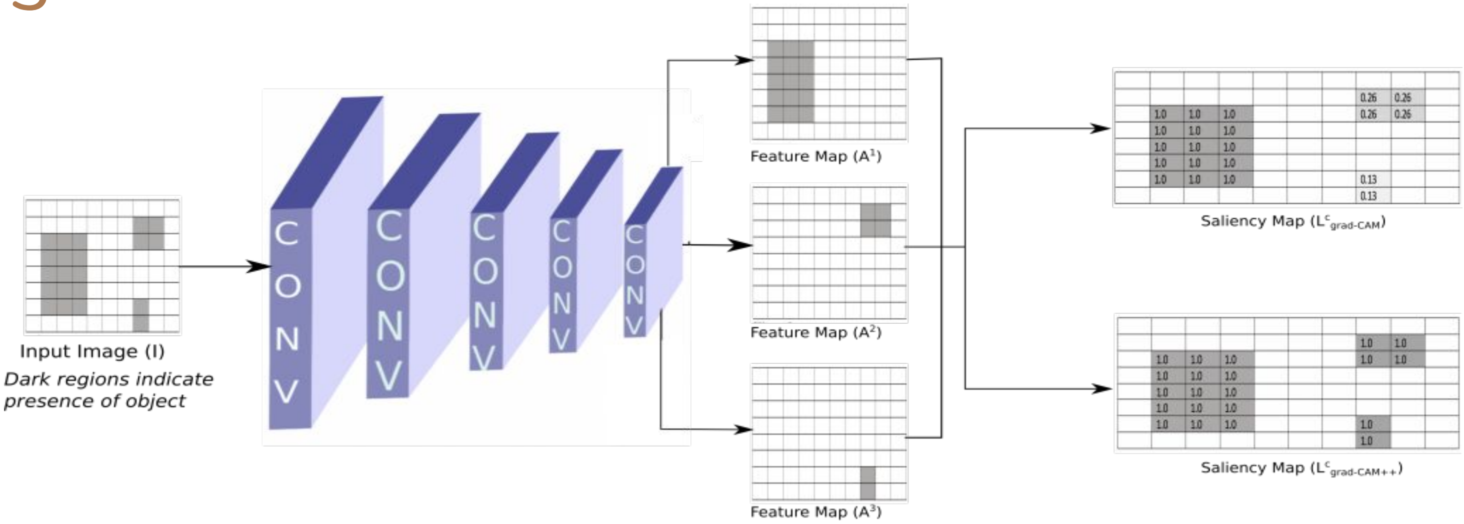


$$L^{c}_{Grad-CAM} = ReLU\left(\sum_{k} w_{k}{}^{c} A^{k}\right)$$

Limitations :

1) When there are multiple occurrences of the target class within a single image, the spatial footprint of each of the occurrences is substantially lower.
2) Unsatisfactory localization performance , especially under occlusion .

gradCAM++



Dark regions indicate presence of object

Saliency Map ($L^c_{grad-CAM}$)

							0.26	0.26	
	1.0	1.0	1.0				0.26	0.26	
	1.0	1.0	1.0						
	1.0	1.0	1.0						
	1.0	1.0	1.0						
	1.0	1.0	1.0				0.13		
							0.13		

Saliency Map ($L^c_{grad-CAM++}$)

							1.0	1.0	
	1.0	1.0	1.0				1.0	1.0	
	1.0	1.0	1.0						
	1.0	1.0	1.0						
	1.0	1.0	1.0						
	1.0	1.0	1.0				1.0		
							1.0		

Dark regions indicate detection of abstract visual features

$$L^c_{Grad-CAM++} = ReLU\left(\sum_k {w_k}^c A^k\right)$$

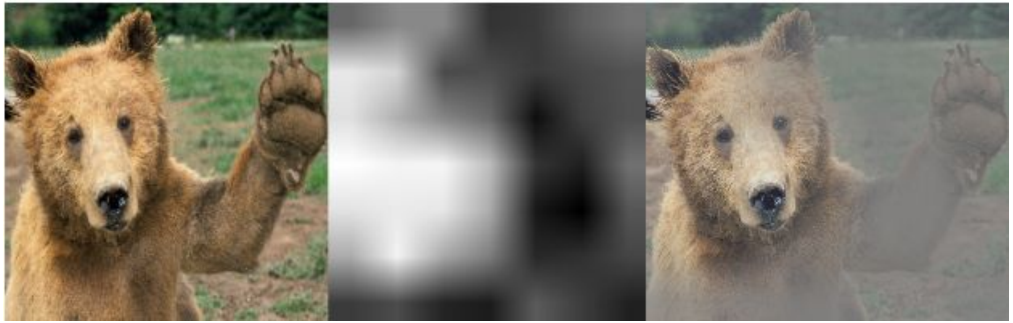
$$where,\ {w_k}^c = \sum_i \sum_j \alpha^{kc}_{ij} ReLU\left(\frac{\partial y^c}{\partial {A_{ij}}^k}\right)$$

Where ,

$$\alpha^{kc}_{ij} = \frac{\frac{\partial^2 y^c}{(\partial {A_{ij}}^k)^2}}{2.\frac{\partial^2 y^c}{(\partial {A_{ij}}^k)^2} + \sum_a \sum_b {A_{ab}}^k \frac{\partial^3 y^c}{(\partial {A_{ij}}^k)^3}}$$

Experimental Setup

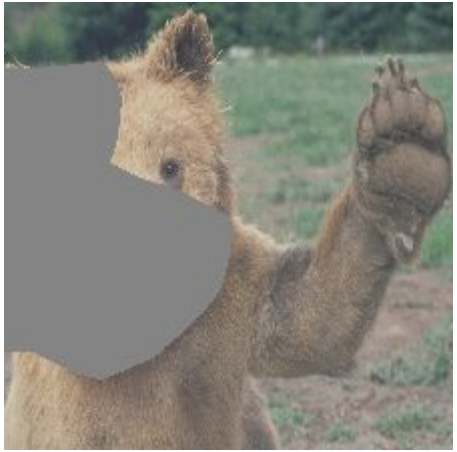
- **METHOD 1 :** Evaluating the explanations by using them to pertubate the image and predicting again to measure the change in confidence value .
- You multiply the original image by the activation map(point wise product). Only regions that score high will still be visible as show below :



- Then you run the new modified "dark" image through the model, and check the new category scores.
- Measure the Increase in confidence(Larger value is better) because the CAM reduced noise from other parts of the image and retains the information that triggers the category output.

METHOD 2 : Completely remove the highest scoring 25%, and see the how much model confidence drops now .

Now this time the more negative is the value the better will be our CAM method , because we are removing the important portion from the image(according to CAM model) and then calculating the model accuracy , so if the CAM has really predicted that what is important so confidence value must drop significantly . This works better than method 1 .

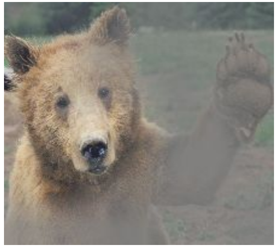


Specifications :

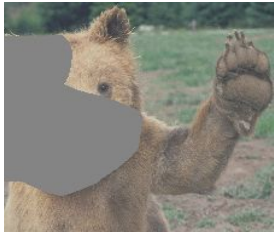
Model : Resnet50 Model Pretrained on ImageNet DataSet which consists of 1000 different Classes . Number of Features at the end of last Convolution layers are 2048 . Performed the Experiments on Class No. 295 of Image Net (American black bear, black bear, Ursus americanus, Euarctos americanus) .

Results :

gradCAM

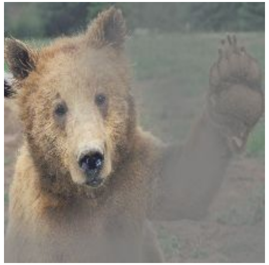


The change in confidence value : 0.0050742062740027905

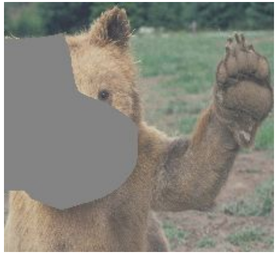


The change in confidence value : -0.001522660139016807

gradCAM++



The change in confidence value : -0.06046486087143421



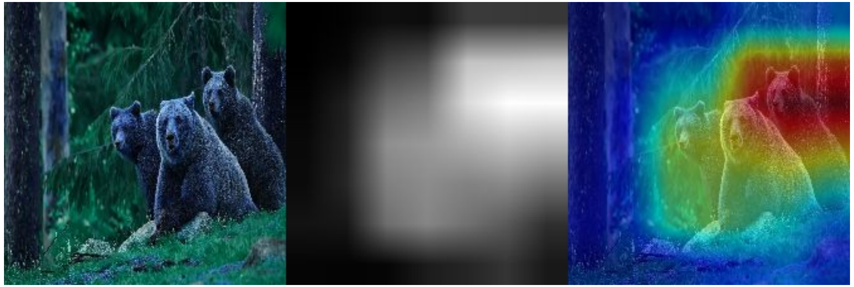
The change in confidence value : -0.0015253901947289705

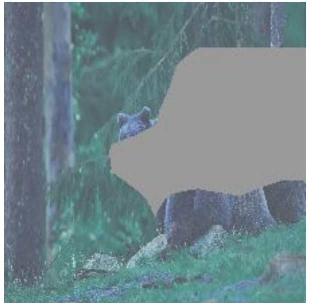
We have performed the Experiments over around 50-60 images of Bear Class and then have taken the average value of confidence change . Because we can’t conclude something on basis of single image experiment .

Results : For gradCAM the average percentage drop is around -9.37 % and for gradCAM++ it is -9.42 %.

Let’s See what happens when we have more than one object of same class :

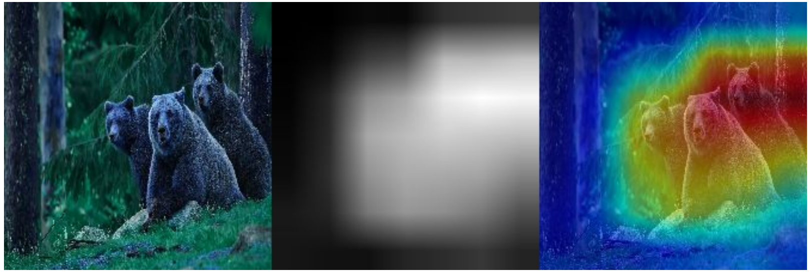
GradCAM

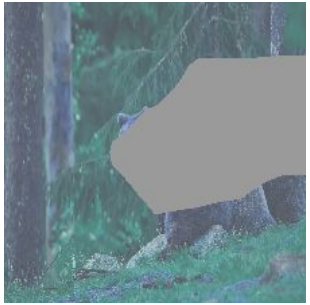




```
The Change in
Confidence Value :
-0.07547037303447723
```

GradCAM++

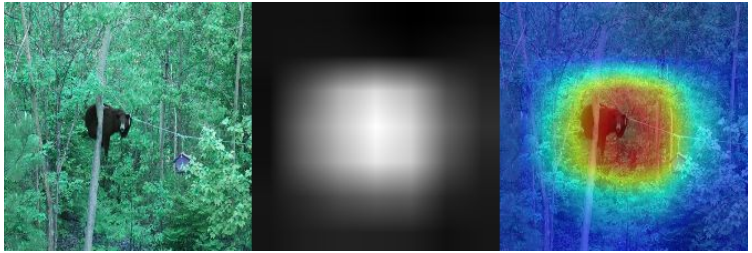




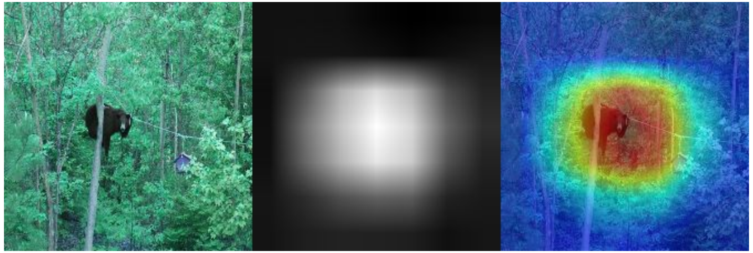
```
The Change in
Confidence Value :
-0.08164670318365097
```

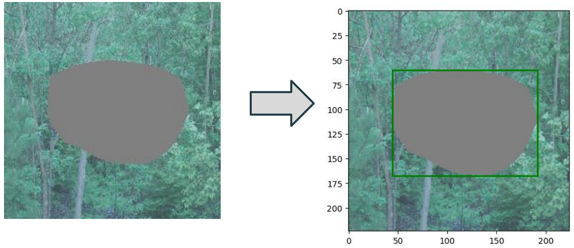
METHOD 3 : Find IoU of Bounding Box enclosing top 25% pixels according to CAM and ground truth bounding box .

gradCAM

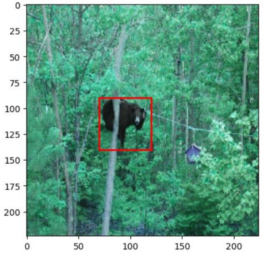


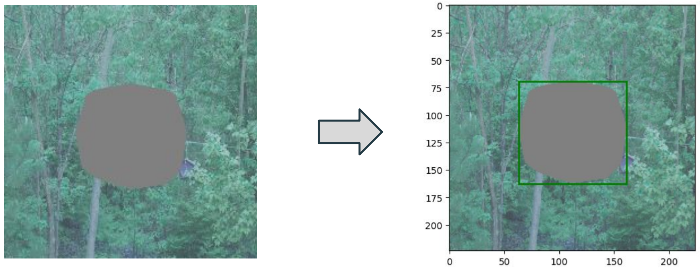
gradCAM++



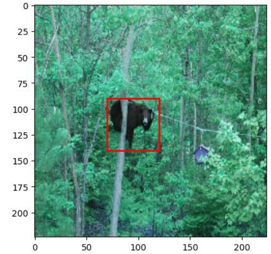


IoU value between two boxes is 0.3951473136915078





IoU value between two boxes is 0.47130228262303736



Conclusions

- gradCAM++ outperforms the CAM and gradCAM models in case where there are multiple objects presence in same image .
- Although there was no significance difference between gradCAM++ and gradCAM for the case of images having single object .
- Explainability in CNN’s are widely used for many applications and hence need the method to evaluate them too .
- The method 1 was not performing well , for some cases the classification score decrease , this may be due to the fact that CAM explanation wasn't telling us the whole story in the first place, and there are other parts that were important as well that is was missing.
- Method 3 also have some implementation problem and hence we move forward with method 2 to evaluate two models on more images . Results show that gradCAM++ performing slightly better than gradCAM .