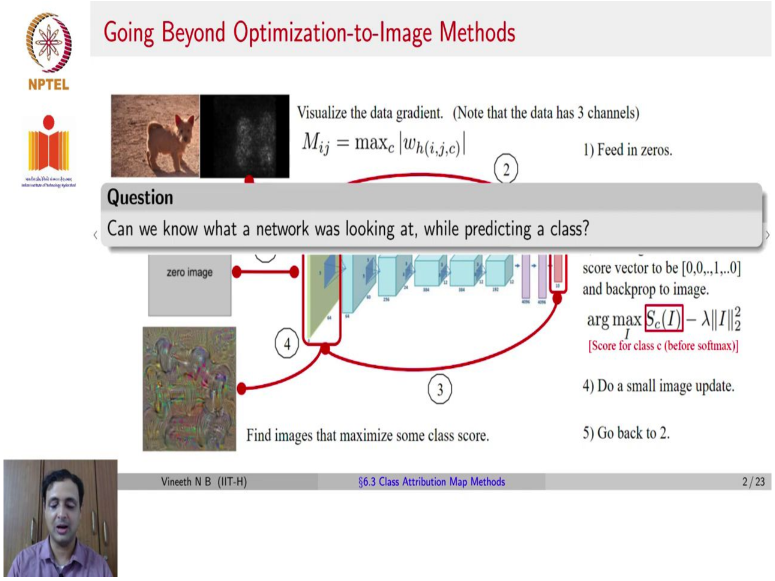
Deep Learning for Computer Vision
Professor Vineeth N Balasubramanian
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad
Lecture 40
Explaining CNNs: Class Attribution Map Methods

Having seen a few different methods that use various aspects of a CNN model to explain its predictions such as its activations of different layers, its gradients, its output probabilities so on and so forth. We will now look at what are known as class discriminative saliency maps or class discriminative attribution maps. By saliency maps we mean maps or regions in an image that are salient for a given prediction.

And by class discriminative saliency map, we mean a saliency map that helps distinguish one class from another, for example if we had a cat and a dog in an image which part of the image led to it being predicted to be a cat and which part of the same image led to it being predicted as a dog. Let us see this in more detail over the next few slides.

(Refer Slide Time: 1:19)



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?

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)]

4) Do a small image update.

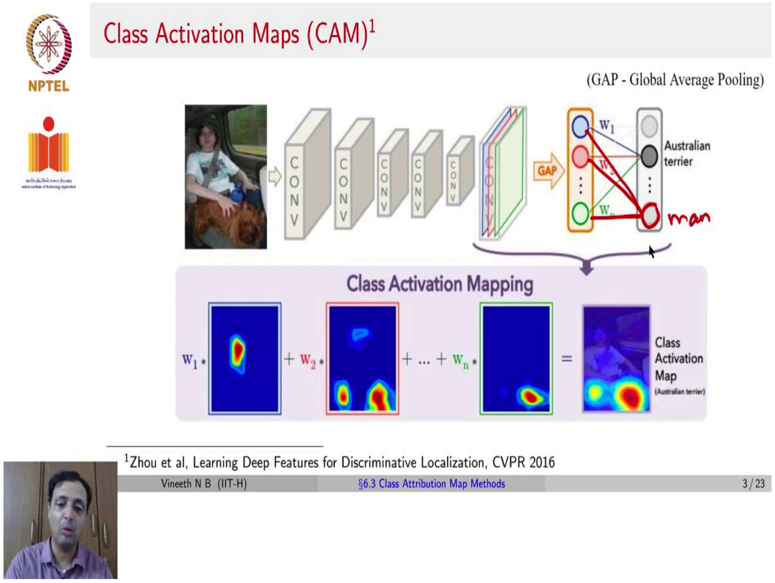
Find images that maximize some class score.

5) Go back to 2.

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 2 / 23

So the question, we still continue to ask is can we know what a network was looking at while predicting a class? But the approach as you will see is different from what we have seen so far.

(Refer Slide Time: 1:35)



Class Activation Maps (CAM)[1]

(GAP - Global Average Pooling)

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

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 3 / 23

One of the earliest methods in this regard was known as class activation maps or cam which was published in 2015 and 16. So this takes a convolutional neural network and uses the notion of global average pooling to achieve the objective. So let us look at this in more detail. So if you had say five to six convolutional layers in your architecture, you take the last convolutional layer.

And then for each map in that convolutional layer remember you could have 100 filters, you would have 100 maps so for each of those maps you do global average pooling or cam. What does global average pooling do? You take a particular attribution map or a feature map sorry not an attribution map a feature map this green one and you average all the intensity values there into one single scalar and that becomes this green circle here.

Similarly, you take the red feature map and average all its values and it becomes the red scalar here. You take the blue feature map average all its values and it becomes the blue scalar here. Global average, it takes the average of the entire image. Now what do we do with these global averages? Having these global averages so each of these scalars here represents one feature map.

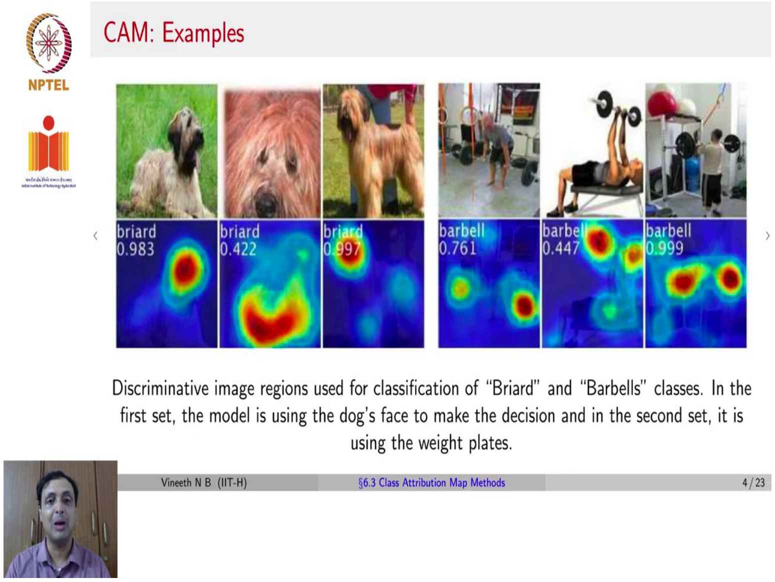
Now, we learn a simple regression model or a linear model that takes us from these scalars to each of the class labels on the last layer. So for each of the class labels in the last layer we learned a w1 into the first attribution into the first feature maps average plus w2 into the second feature maps average and so on and so forth until wn where n is the number of feature maps in that last convolutional layer.

Now, how does this help us? So for a given image such as the image that you see here, if you forward propagate address through a trained AlexNet, you would get a set of n activation maps as the at the last convolutional layer. Let us assume that these are those activation maps so you see one here, the second one here and so on and so forth until the last one here. Now, the weights that we learned between the output of the gap layer and the classification layer are now used to weight each of these activation maps.

And when you do a weighted sum of all of these activation maps that gives us the contribution of these activation maps towards one particular class label. In this example, let us say we want to predict this image as belonging to an Australian terrier. Then you learn weights corresponding to each of those activation maps to the Australian interior class and you weight each activation map in the same way.

And this weighted combination of activation maps of that conv5 layer corresponds to the Australian terrier. If you say you wanted to predict a man in the image through this then you would have a different set of weights maybe let us say this was the man class so then you would have a different set of weights of the same activation maps that connect you to the man class. So a different weighted combination of the same activation maps will tell you which part of the image corresponded to the man class. Now, this approach gives us a way of getting us class discriminative saliency maps.

(Refer Slide Time: 5:36)



CAM: Examples

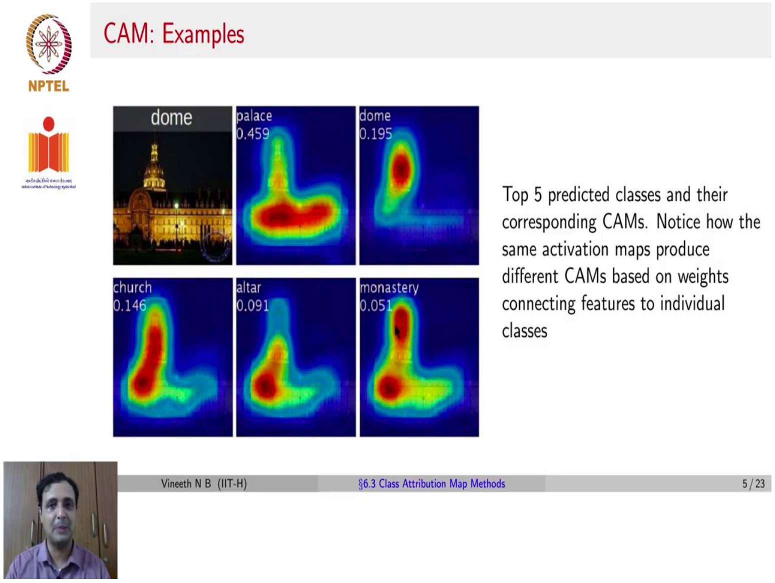
Discriminative image regions used for classification of "Briard" and "Barbells" classes. In the first set, the model is using the dog's face to make the decision and in the second set, it is using the weight plates.

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 4 / 23

There are some advantages and disadvantages as we will see soon but here are some visual examples to start. So you can see here that in the first three images it is examples of the class Briard and the second three examples it is the class barbells. And you can see here that in each of these cases the CNN model looks at this particular region and calls it a Briard.

Similarly so in this case similarly so in this case and similarly for the barbells you can actually see that the CNN looks at the weight plates to make the decision that this image corresponds to barbells.

(Refer Slide Time: 6:22)



CAM: Examples

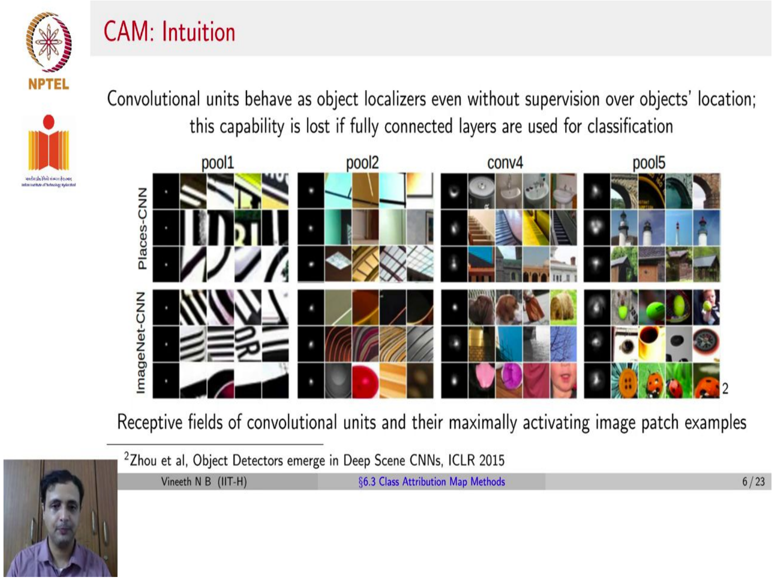
Top 5 predicted classes and their corresponding CAMs. Notice how the same activation maps produce different CAMs based on weights connecting features to individual classes

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 5 / 23

So obviously you can ask the question: what if I change the class? So let us see an example for that too. So here is an image and here are the predicted saliency maps or class activation maps for the top five predicted classes based on this image. The top five predicted classes were palace, dome, church, altar and monastery and you see here that when the model was predicting the palace it was looking at the entire structure.

When it was predicting domes it was only looking at the dome. When it was predicting church it looked at only the facade close to the dome and similarly for the altar and for the monastery it was looking at certain parts which probably made it think it was a monastery. So this gives us as we just mentioned class specific activation maps which can be useful in practice.

(Refer Slide Time: 7:24)



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]

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

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 6 / 23

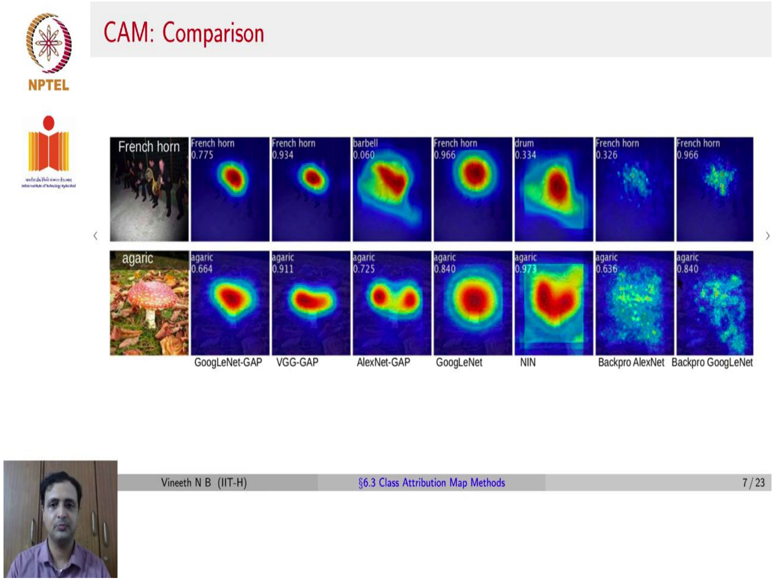
An intuition for doing cam is that in any CNN so far that we have seen in most CNNs we have had a few convolutional layers, then followed by some fully connected layers. Convolutional layers do maintain a certain level of object localization capability. If you recall in the first lecture we saw an example of the conv5 feature map with two people sitting in an image.

And we saw that the feature map actually showed where they were positioned in the image. So the convolutional layer did give us an idea of where the objects were localized with no supervision or with no details given to us explicitly. However, if you try to look at the representation that you get after a fully connected layer for example the fc7 representation, you would lose this information.

That is the very nature of the difference between a convolutional layer and a fully connected layer. So that is one of the reasons for developing a convolutional layer itself. So that is now by doing the cam based approach. We actually see now that we can retrieve back the activation maps of the fifth layer and be able to use them to explain our decisions at the classification layer.

These are just more examples here of the localization capability of CNNs in different feature maps and you can see here that you have different receptor fields of convolutional units and the patches that maximally activate that patch. And you can see here that you do get a certain set you can you get a certain sense of localization through these kinds of feature maps at different convolutional layers. You obviously lose that when you go to fully connected layers.

(Refer Slide Time: 9:21)



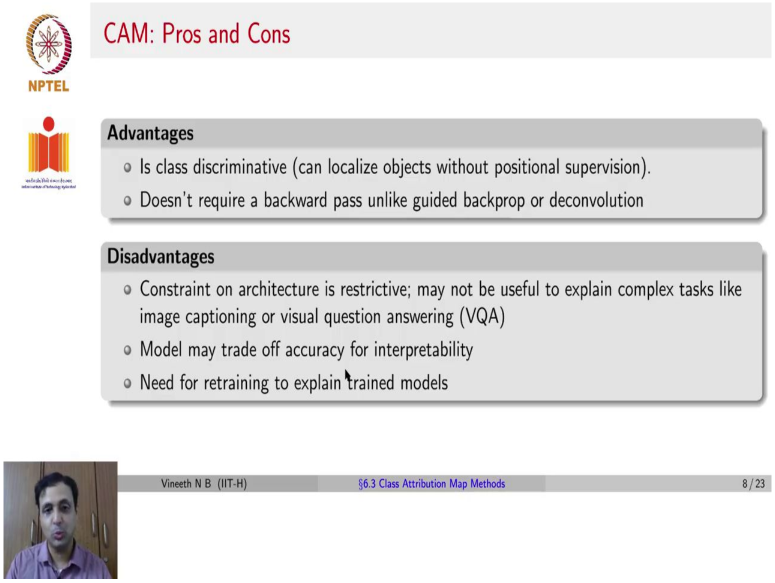
CAM: Comparison

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 7 / 23

Here are a comparison of cam maps across different models. You can see here this is applying cam or using gap global average pooling on GoogleNet, vgg, AlexNet just GoogleNet alone another architecture known as NIN or network in network and these are compared with a back propagation on AlexNet and a back propagation on GoogleNet.

Back propagation is your data gradient remember we talked about the data gradient in the earlier lecture. So this is that visualization and you can clearly see that cam gives a far stronger saliency map a far more useful saliency map when compared to the data gradient.

(Refer Slide Time: 10:09)



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

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 8 / 23

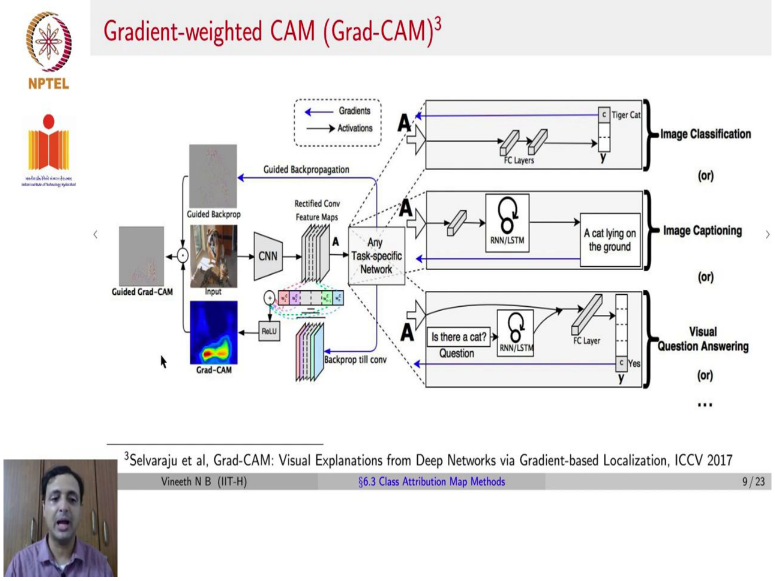
What are the pros and cons of cam? Can you think about it? Can you think of any significant con disadvantage of cam? Advantages class discriminative can localize objects without any positional supervision and it does not require a backward pass through the entire model unlike something like guided back prop or back prop to image. What are the disadvantages? The disadvantage is that you actually the key one is actually the third bullet here which is there is a need to retrain these models to be able to get those weights that we get after global average pooling.

After training an AlexNet you still have to do global average pooling and learn those many linear models at the last layer to be able to understand the relationship between the activation maps and each class label. You will have to do this retraining explicitly for explanations on top of training your AlexNet or any other CNN model and that can become an additional computational burden.

And we are imposing a constraint on architecture by saying that you will have to introduce a global average pooling layer to be able to explain your model. And that may cause problems. When you want to generalize to many vision tasks using this kind of a method and there is a chance that the model may trade off accuracy for interpretability. So to get a better interpretability it may end up achieving lower accuracy.

If you used the gap model and the corresponding weights itself for classification. Now let us try to try to see how we can address these disadvantages which was done in a follow-up method called grad cam which is published in ICCV of 2017.

(Refer Slide Time: 12:05)



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

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

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 9 / 23

ICCV is a top tier computer vision conference. Grad cam stands for gradient weighted cam. As we will see in the next few slides it is a very intelligent approach on repurposing cam using existing quantities in a CNN and let us see how that works.

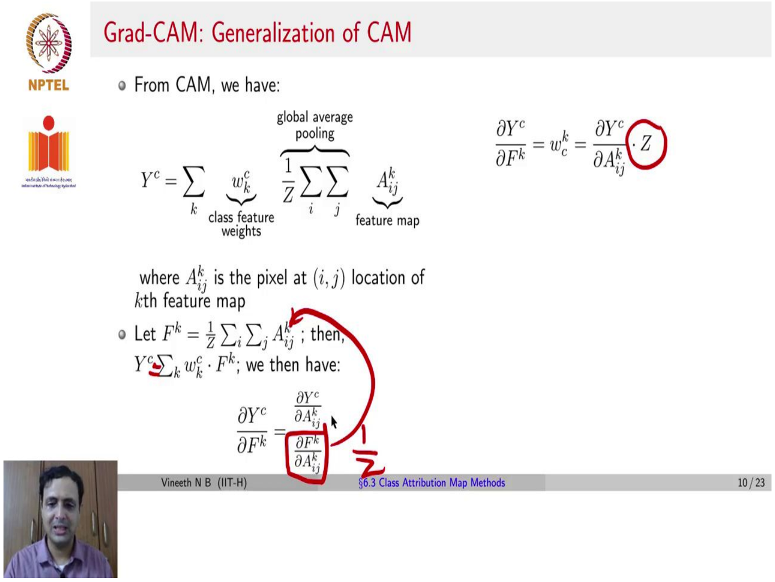
But here is the overall idea and architecture we will describe each of these components over the next few slides. So you have the input image here you send it through a CNN. You have your convolutional feature maps and the convolutional feature maps could be followed by any task specific network.

You could be doing classification which is what we have seen so far but you could also use their approach for any other tasks such as image captioning or visual question and answering. These are tasks that we will see later in this course, irrespective of the task we assume now that there is a last layer there is a loss and there is a gradient which can be assumed in any neural network for that matter.

Once you have the gradient of a loss with respect to any task you have now gradients with respect to all of the feature maps, so the activation maps. You now combine them you combine the gradients that you get for each of those activation maps and they automatically become your weights for each of the feature maps.

And in grad cam because we want to ensure that only positive correlations are shown in the final saliency map we apply a ReLU on the on the weighted combination of the activation maps and that becomes our final grad cam saliency map. The method also talks about adding guided back propagation to make a variant of grad cam called guided grad cam. Let us see this in a bit more detail and also mathematically as to why grad cam becomes an extension of cam.

(Refer Slide Time: 14:08)



Grad-CAM: Generalization of CAM

- From CAM, we have:

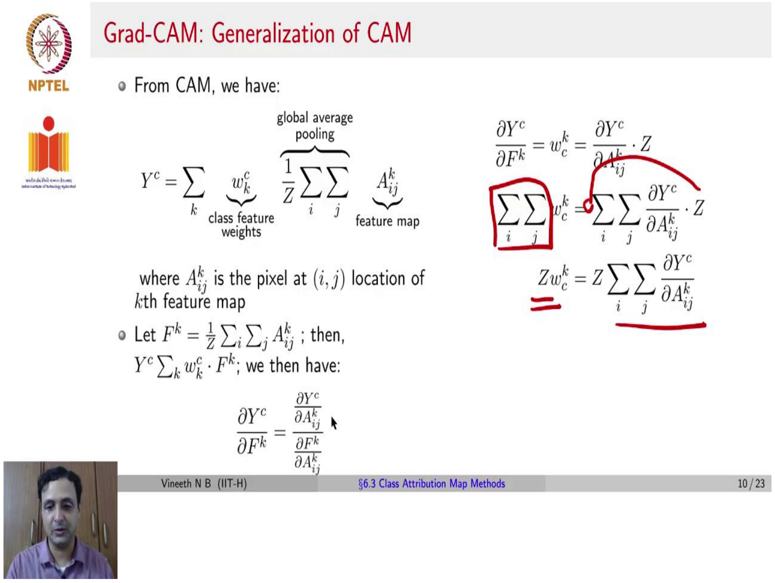
$$Y^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}$$

where A_{ij}^k is the pixel at (i, j) location of kth 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:

$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A_{ij}^k}}{\boxed{\frac{\partial F^k}{\partial A_{ij}^k}}} \quad \frac{1}{Z}$$

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



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 kth feature map

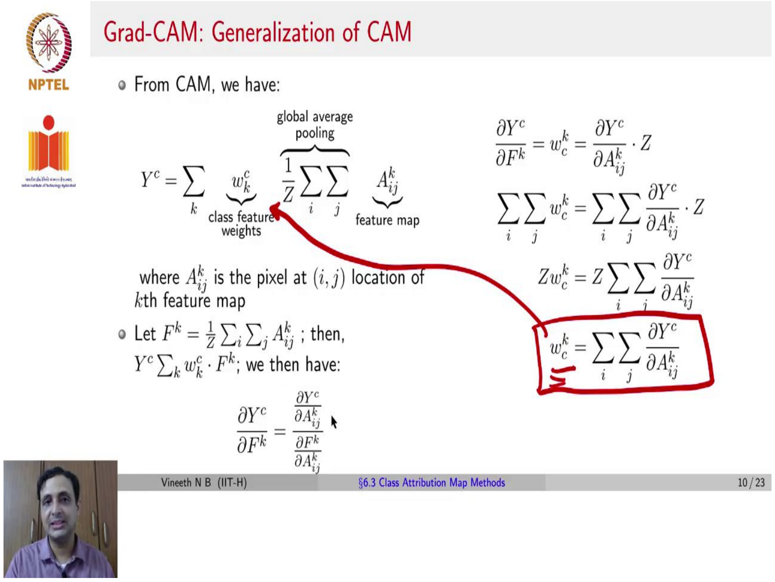
- Let $F^k = \frac{1}{Z}\sum_i\sum_j A_{ij}^k$; then, $Y^c \sum_k w_k^c \cdot F^k$; we then have:

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

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

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

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



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 kth 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:

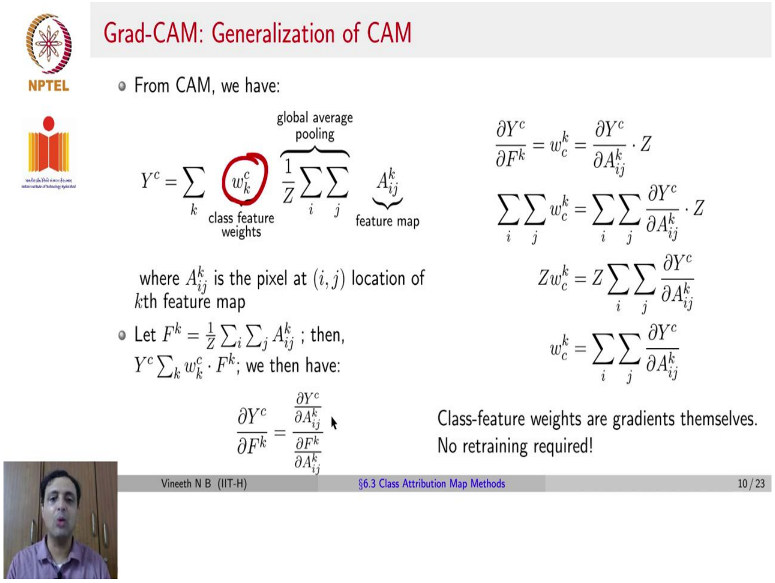
$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A_{ij}^k}}{\frac{\partial F^k}{\partial A_{ij}^k}}$$

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

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

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

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



Grad-CAM: Generalization of CAM

- From CAM, we have:

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

where A_{ij}^k is the pixel at (i,j) location of kth 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!

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 10 / 23

From cam mathematically speaking, we have Yc which is your scores or the class course in the last layer which is given by summation over k which is all your k feature maps. We assume now that you have k such feature maps and you have a class weights for each of these feature maps and summation over i summation over j, a i j k, 1 by z is going to be your global average pooling of each of the k feature maps.

$$Y^c = \sum_k w_k{}^c \tfrac{1}{Z} \sum_i \sum_j A^k{}_{ij}$$

And then you have the weights and in cam we know that these wc case wc corresponds to the weight for each class and wk corresponds to the weight for each activation map. So you need to do both. This is what we learn through linear models in that last layer. Now let us go from there. Let us now assume that f superscript k is given by the last terms 1 by z summation over i summation over j, a i j superscript k.

$$F^k = \tfrac{1}{Z} \sum_i \sum_j A^k{}_{ij}$$

Then you are going to have Yc is going to be given by sorry there is an equal to missing there so you are going to have Yc to be equal to summation over k wck into fk. We are simply replacing the last terms with fk and if you now took the gradient of Yc with respect to fk, you have dou Yc by dou fk is equal to dou Yc with respect to dou a i j k divided by dou fk with respect to dou a i j k.

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

We just have the same component that we are taking the derivative with respect to. If you look at this closely dou Yc by dou fk which is what wc is we can see that from this equation here. Remember once again that this Yc is equal to summation wck fk. So which means dou Yc by dou fk will be wck for a particular k whichever fk you chose for a particular feature map. And that is given by dou Yc with respect to dou a i j k divided by dou fk by dou a i j k.

$$\frac{\partial Y^c}{\partial F^k} = w^k{}_c = \frac{\partial Y^c}{\partial A^k{}_{ij}} Z$$

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

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

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

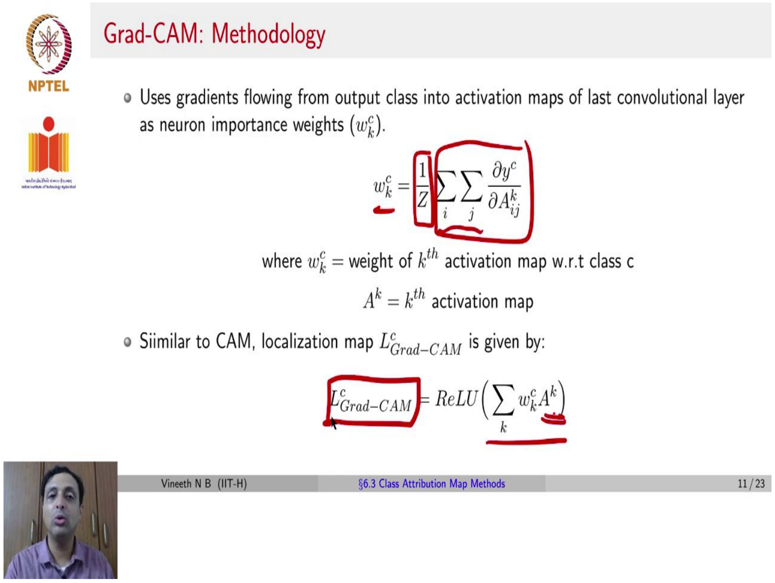
Now, dou fk by dou a i j k by the very definition of f j will turn out to be 1 by z and because that is 1 by z in the denominator you get an into z on the numerator here when you when you write this out more clearly. So what does this tell us? Now if we sum the terms on the left hand side over i and j which are all your pixel locations in each feature map, you similarly have a summation on the right hand side.

Now the summation over inj for wck because wck does not depend on i and j is just going to be z which is the total number of pixels in each feature map or activation map and similarly the z constant comes out here and you have your summation over i summation over j dou Yc by dou a i j k here. Rather we can say wck then is given by summation over i summation over j dou Yc by dou a i j k.

This tells us tells us something important that the wck which we actually learned in the cam model are actually simply the summation of the gradients of each the each of the class score with respect to every pixel in the feature map and adding them all up. So in truth the class feature weights here are the gradients themselves and we actually do not need to do the retraining the way we saw it with cam.

You do not need the global average pooling you do not need the retraining those weights that you did the global average pooling for can actually be obtained as gradients of the last layer scores with respect to any feature map or activation map with which you want to compute your saliency maps.

(Refer Slide Time: 18:35)



Grad-CAM: Methodology

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

$$\underline{w_k^c} = \boxed{\frac{1}{Z}}\boxed{\underline{\sum_i\sum_j} \frac{\partial y^c}{\partial A^k_{ij}}}$$

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 $L^c_{Grad-CAM}$ is given by:

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

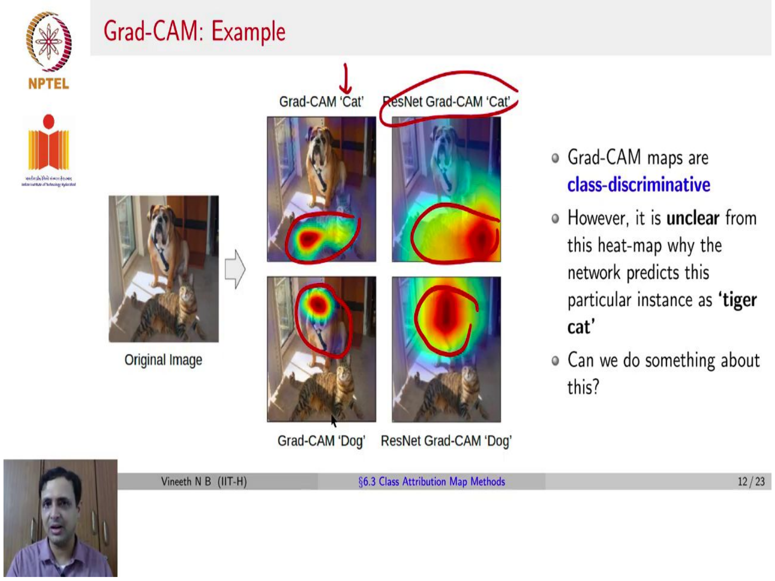
Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 11 / 23

So which means we can now write out wck as summation over i summation over j dou Yc by dou a i j k. We are going to have a normalization factor 1 by z because we want to divide it across all of the pixels.

$$w_k{}^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^k{}_{ij}}$$

This becomes our new weight and our final saliency map or the localization map in grad cam becomes given by summation over k wck ak where ak are the different k activation maps that we have. And because we want only the positive correlations to be shown on the final saliency map we apply a ReLU on it to get the final to get the final image.

(Refer Slide Time: 19:18)



Grad-CAM: Example

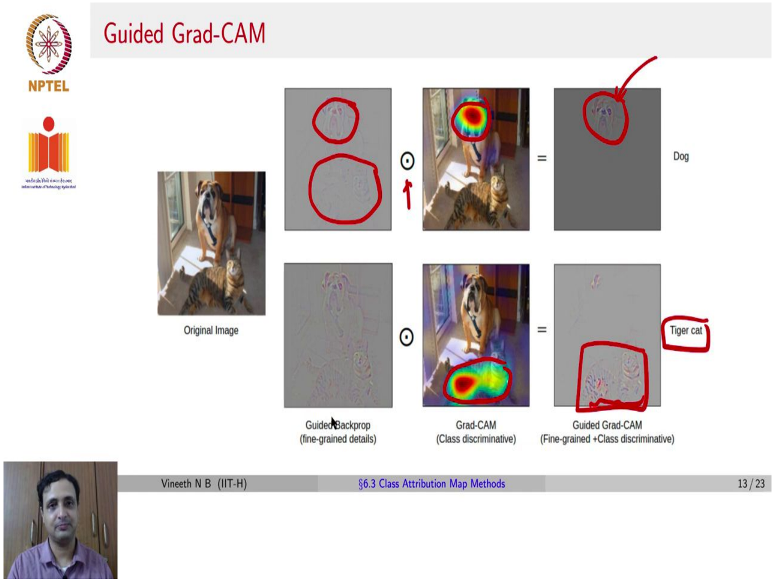
- Grad-CAM maps are **class-discriminative**
- However, it is **unclear** from this heat-map why the network predicts this particular instance as **'tiger cat'**
- Can we do something about this?

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 12 / 23

You see examples now for original image here is the grad cam for cat which focuses on the cat and similarly doing this with the ResNet model, here is the location of the cat and similarly for a dog it looks at the dog to be able to say it is a dog and similarly for the ResNet model looks at the entire dog to be able to say it is a dog, which is quite good. But one careful observation here is that the network actually predicts this to be a tiger cat and not just a cat.

So can we elaborate on this further and see why it is a tiger cat and not just a cat, can we try to do anything further?

(Refer Slide Time: 20:02)



Guided Grad-CAM

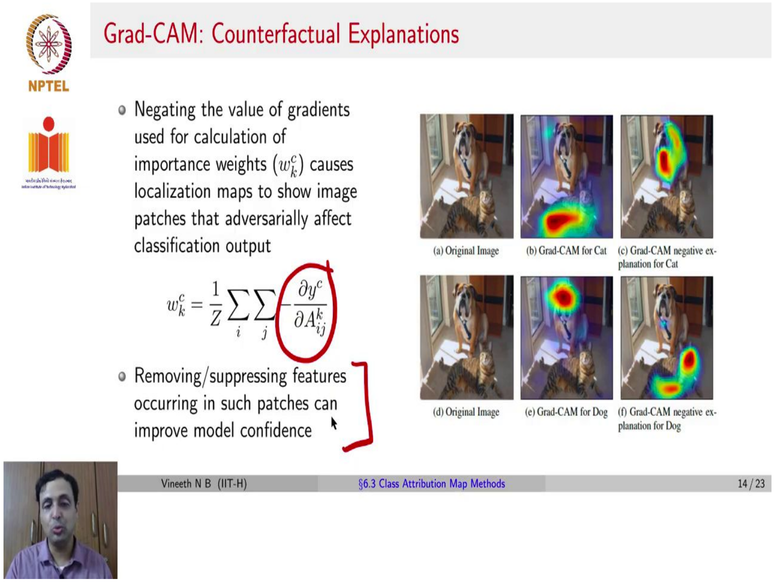
Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 13 / 23

And here is the where here is where the method now proposes a variant of grad cam called guided grad cam which brings together guided back propagation that we saw in the earlier lecture and grad cam together and juxtaposing them one on top of the other by doing what is known as a hadamard product or a pixel wise product.

And we now see that taking this particular region which was pointed out by grad cam and combining it with guided back propagation output in terms of what was salient in the image, we now get a clearer estimate of what was the dog. Remember guided back propagation was not necessarily class discriminative. So if you only used that you would also have other kinds of pixels which are active as you can see here.

But by combining it with grad cam, we get a more localized understanding of what the CNN was looking at while calling it a dog. And now, you see why you called it a tiger cat why the model called it a tiger cat because you take the grad cam saliency map and combine it with guided back props output again and you actually get this kind of a output which shows the striations on the body of the cat which explains why it was a tiger cat.

(Refer Slide Time: 21:27)



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

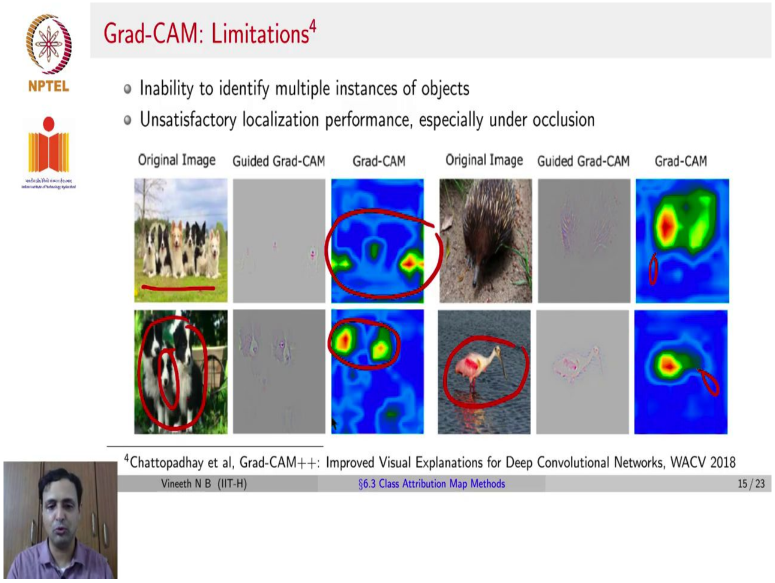
Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 14 / 23

Grad cam went on to also show how this method could be used for what are known as counterfactual explanations. Rather can you try to find out which class or which region in the image maximally affected my model in calling an image as belonging to a dog. So I have an image I want to label it as a dog and the model does give me certain probability of the label being a dog.

But can I find out which other pixels in the image may have suppressed my probability for a dog as the output of the model and this can be done by using the exact same grad cam procedure. The only difference now is use negative gradients instead of the positive gradients and this would tell us which combination of feature maps or which weighted combination of feature maps were negatively influenced a particular class probability to be high.

And that would give us what are known as counter factual explanations and eventually how do you use this? You can remove or suppress these features in some way to improve the confidence of the model if you would like to use it in that particular way.

(Refer Slide Time: 22:55)



Grad-CAM: Limitations[4]

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

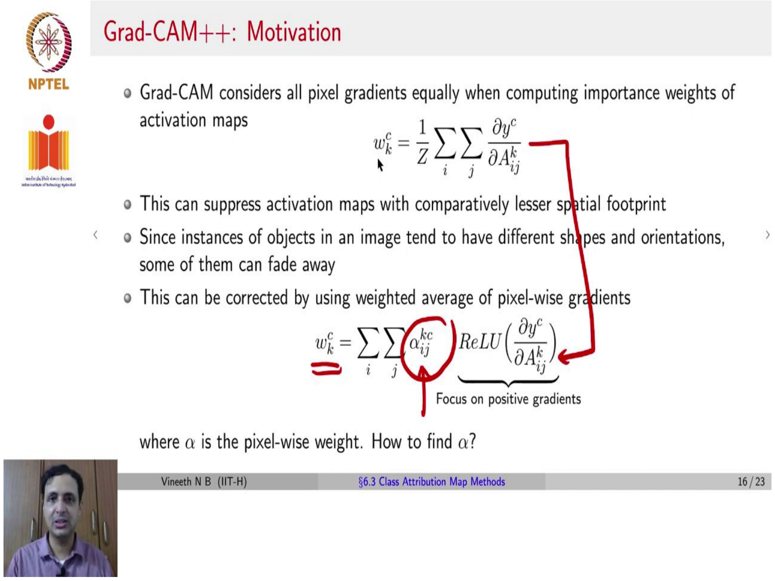
[4] Chattopadhay et al, Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks, WACV 2018

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 15 / 23

Grad cam however had some limitations when there were multiple instances of objects or when there were occlusions in the image. So here is an image where there are multiple dogs and you can see that grad cam gives such an output where it does not seem to capture all the dogs in the image, maybe it does somewhat well when there are fewer number of objects there are still three dogs here and grad cam captures two dogs but misses the one in the middle that is one of the limitations of grad cam. Another case where it seems to not completely get a good saliency map is where there are occlusions.

You see a bird here whose legs are hidden underneath the water and as well it has a beak and here you see a hedgehog which has a beak-like structure too. In both of these cases grad cam does not seem to capture those aspects which are actually salient aspects of that object as class discriminative in its in its visualization. Can we do something about it to our or are there any limitations in the formulation of grad cam itself that we can improve.

(Refer Slide Time: 24:13)



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 α is the pixel-wise weight. How to find α?

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 16 / 23

And this was done in a work called grad cam plus plus and the main motivation of grad cam plus plus is observing that grad cam took and took the gradients of dou Yc with respect to dou, each of the pixels in your activation maps and then took an average of all of them to get its final weight. In a sense it is weighting each pixel equally by taking the average when it when it gets the final weight.

Grad cam plus plus’s idea states that maybe pixels that contribute more towards the class should get more weight than have equal distribution while computing this weight w k c. Let us see how we can do that. So this can especially suppress activation maps with lesser spatial footprint rather we saw this example on the previous slide too when you have three dogs and there is one dog which is smaller the other two dogs got most of the gradient and the third dog did not because it has a lesser spatial footprint.

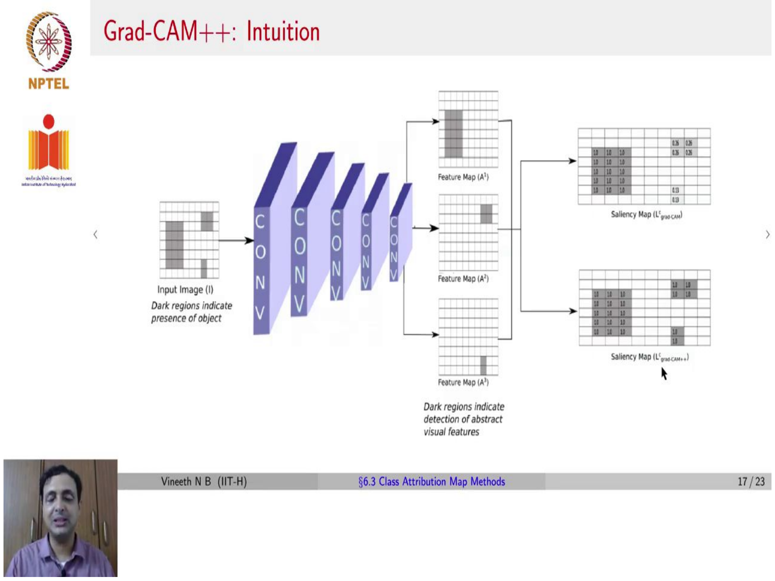
We will see this more clearly on the next slide and when you have this kind of a bias in the visualization some of the smaller objects may just not be picked up in the saliency maps. What can we do about this? Grad cam plus plus suggests that we retain the same formulation of grad of grad cam. However, this time while computing your final weight wkc we are going to give each pixel in each activation map a certain weight as to how they must be how they must contribute to that saliency map.

So let us call those constants alpha sub i j superscript kc so alpha sub i j corresponds to each i jth location of a feature map. K corresponds to the kth feature map and c corresponds to the class which we want to maximize. Grad cam plus plus also adds a ReLU here to ensure that only positive gradients are considered in the computation of this weight. But the larger question here is how do you get these weights?

$$w_k{}^c = \sum_i \sum_j \alpha_{ij}{}^{kc} Relu\left(\frac{\partial y^c}{\partial A^k{}_{ij}}\right)$$

In grad cam, it was simpler to average all of these gradients that you get and use that to get a wkc and remember each wkc then becomes the weight of the kth feature map towards the cth class. So but now how do you compute these alpha ijs at each pixel level?

(Refer Slide Time: 26:44)



Grad-CAM++: Intuition

NPTEL
Input Image (I)
Dark regions indicate presence of object
CONV
CONV
CONV
CONV
CONV
Feature Map (A^1)
Feature Map (A^2)
Feature Map (A^3)
Dark regions indicate detection of abstract visual features
Saliency Map ($L^c_{grad\text{-}CAM}$)
Saliency Map ($L^c_{grad\text{-}CAM++}$)
1.0
0.26
0.13

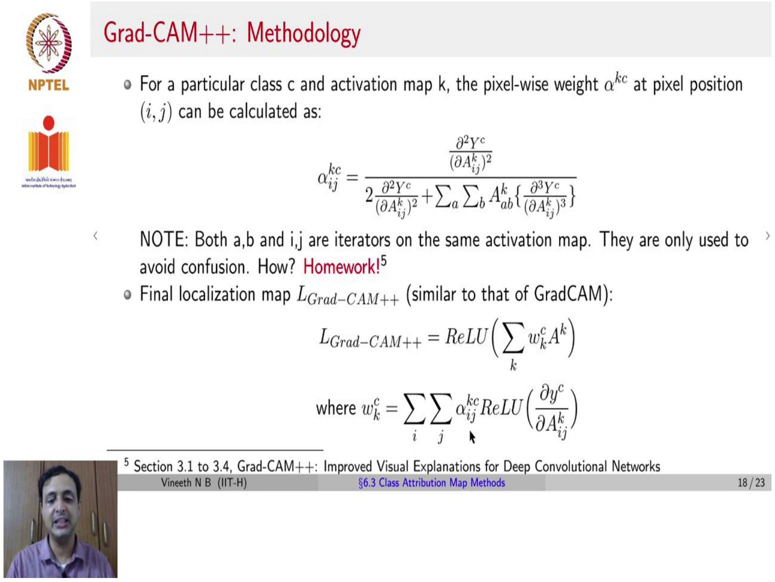
Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 17 / 23

Before we go there, let us try to understand the intuition of grad cam plus plus again visually. So here you notice that if you had an image with three different objects say dogs a dog of a large occupying a larger special footprint, another dog occupying a mid-level special footprint and another dog occupying a small footprint and for the moment let us assume that different feature maps capture different dogs. This for instance could be any other object for that matter could have been a dog, cat and a jug or something like that.

So each feature map let us assume captures each of these objects and you see here that in grad cam when the saliency map gradients are computed you can see that the area with the largest footprint ends up getting most of the gradient, while the gradient towards the rest of the pixels are smaller because they have fewer pixels and hence contribute lesser towards the output while grad cam plus plus tries to overcome this by doing the pixel wise weighting and you can see here that in grad cam plus plus the weights are in the same range those the gradients are in the same range when you use this kind of an approach.

This is actually the final saliency map that is in the same range for grad cam plus plus. So we still are left with the question in grad cam plus plus as to how do you compute those alphas at a pixel level.

(Refer Slide Time: 28:10)



Grad-CAM++: Methodology

- For a particular class c and activation map k, the pixel-wise weight α^{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):

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

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

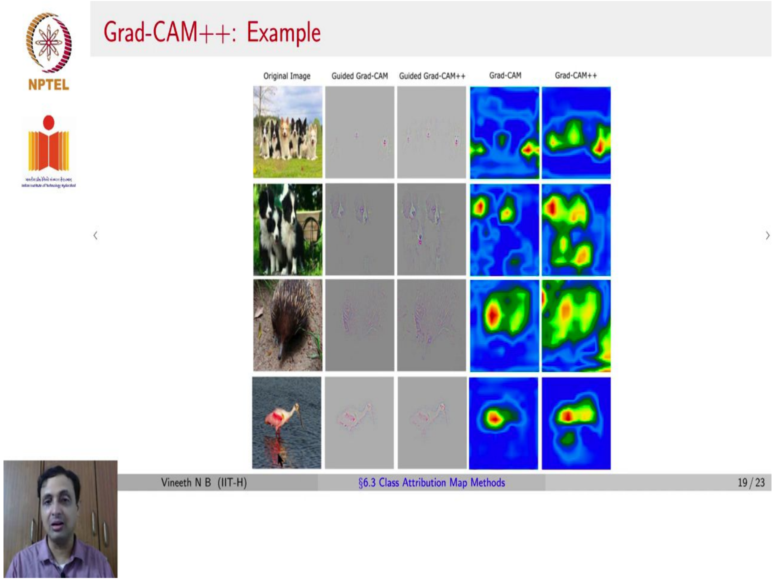
[5] Section 3.1 to 3.4, Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 18 / 23

We are not going to derive this here this can be lengthy and that is going to be part of your homework. But what happens is by reorganizing the gradients and using some arithmetic around the expressions of the gradients, grad cam plus plus shows that $\alpha_{ij}{}^{kc}$ can actually be obtained as a closed form expression of several gradients that you have already with you. Both a b and i j here are iterators on the same activation map and you can go ahead and look at the paper for grad cam plus plus to understand how this derivation is done.

But once this derivation is done, the rest of it stays very similar to grad cam. In grad cam plus plus you still do a ReLU at the end of summation over k wck ak where wck is given by summation over i summation over j alpha i j k c ReLU of dou Yc by dou a i j k. How does this help?

(Refer Slide Time: 29:18)

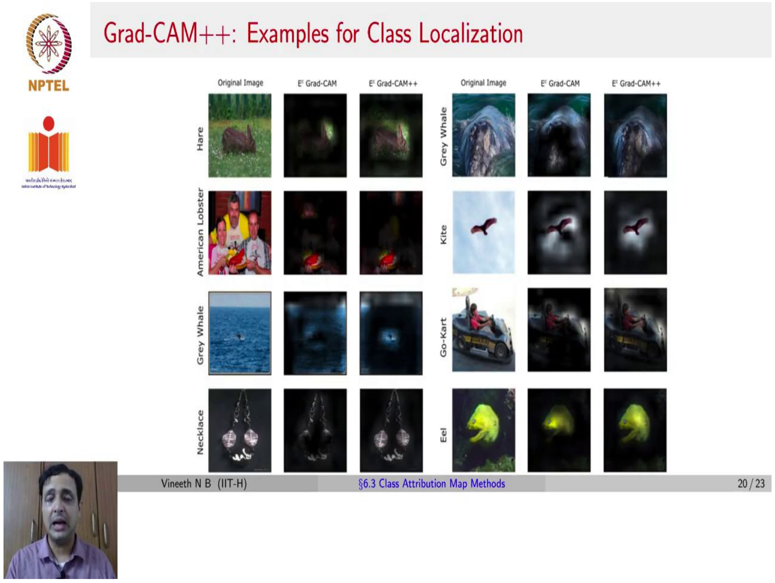


Grad-CAM++: Example

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 19 / 23

You now see that given the same image of multiple dogs while grad camp did not localize all of them effectively. Grad cam plus plus seems to get a better saliency map around the dogs and it works a bit better even when there are three dogs and it also captures the setting when there are structures such as beak or legs under occlusion as compared to grad cam.

(Refer Slide Time: 29:44)


Grad-CAM++: Examples for Class Localization

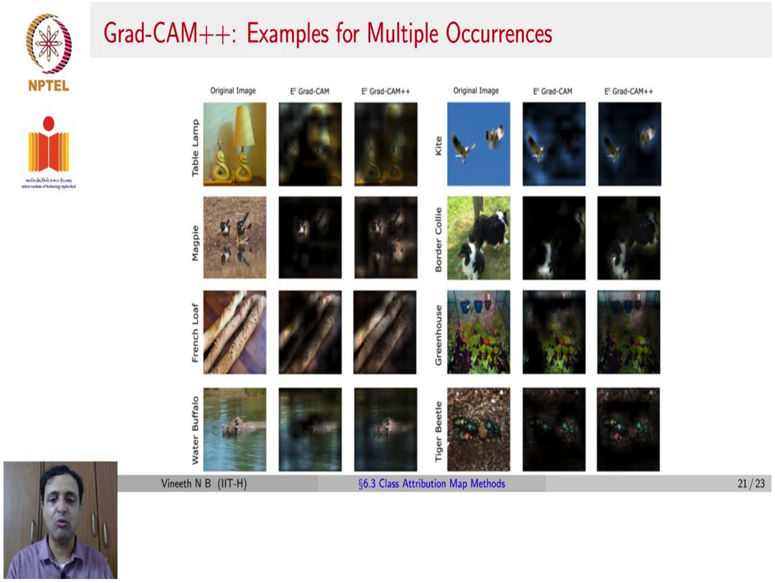
Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 20 / 23

Grad cam plus plus also showed that the saliency maps obtained from grad cam plus plus give better localization if compared to the bounding boxes that are provided with images when compared to grad cam.

And you see several results here the first column in this left block are the original images then you have the corresponding grad cam visualizations for each of these classes Hare, American lobster, gray whale and a necklace and you see the grad cam plus plus localization which seems to be better especially for some things like gray whale when compared to grad cam. You see a similar set of images for another gray whale here.

A kite a go-kart and an eel where grad cam plus plus localizations improve over grad cam by considering the pixel wise waiting strategy.

(Refer Slide Time: 30:37)

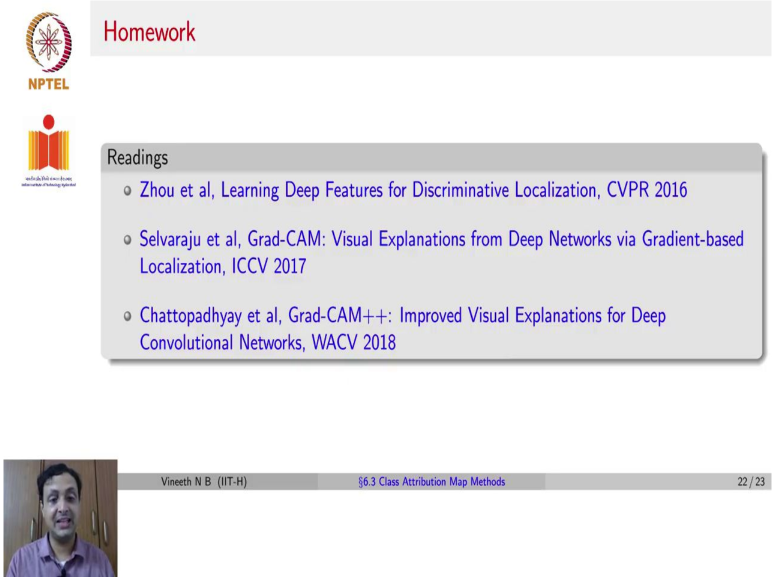


Grad-CAM++: Examples for Multiple Occurrences

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 21 / 23

Here are more examples of grad cam plus plus for multiple occurrences of objects, once again improved performance over grad cam.

(Refer Slide Time: 30:50)



Homework

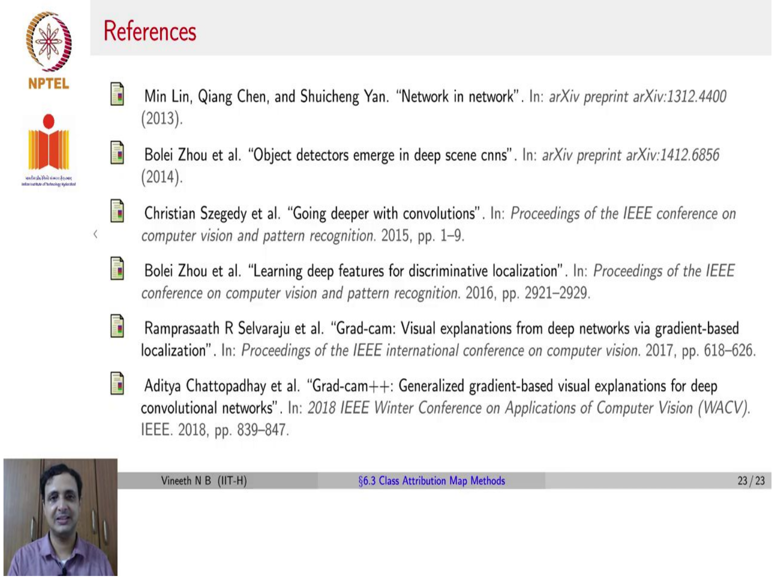
Readings

- 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

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 22 / 23

For homework, there are these three papers cam, grad cam and grad cam plus plus and your job would be to read through them and the other exercise would be to work out how we get the closed form expression for alphas in grad cam plus plus.

(Refer Slide Time: 31:10)



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.

Vineeth N B (IIT-H) §6.3 Class Attribution Map Methods 23 / 23

References.