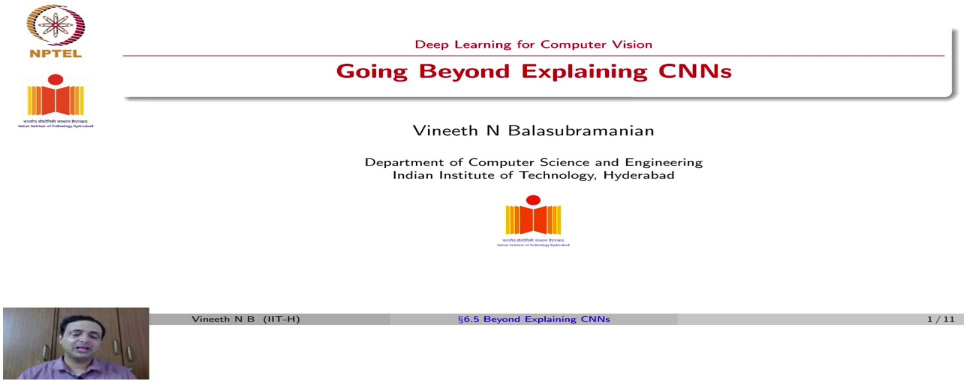


Deep Learning for Computer Vision

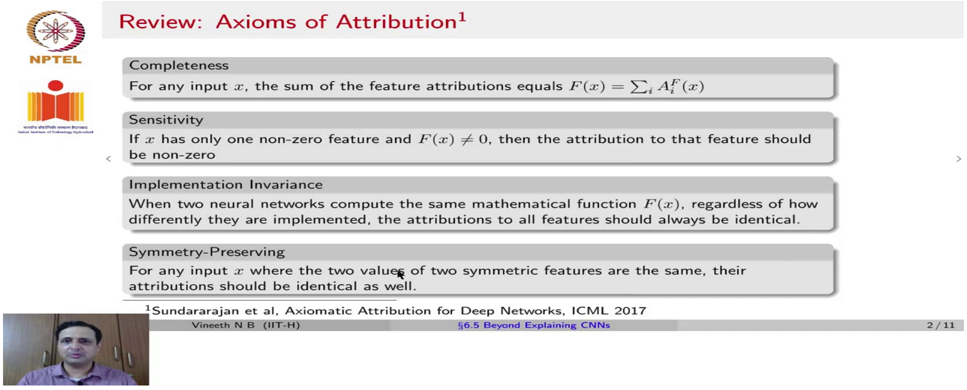
Going Beyond Explaining CNNs

Vineeth N Balasubramanian

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

Vineeth N B (IIT-H) §6.5 Beyond Explaining CNNs 1 / 11

The completeness axiom says that for any input x, the sum of the feature attributions equals F of x, which means if you had say N different attributes or 10 different attributes in your input, the sum of their attributions should be the, the actual output that you get of the neural network, the sum of the attributions respect to your neural network should be the actual output.So let us assume that you had two neural networks, one three layer on one two layer, but it happens at the end, that the exactly give the same output for every input that you can give, by some means they have learned the same function, maybe use Relu in one three layer network and use a Sigmoid in another two layer network.Another axiom states about the symmetry preservation of the attribution method, which states that for any input x, where the two values of two symmetric features are the same, their attributions should be identical as well.



Review: Axioms of Attribution[1]

Completeness
For any input x, the sum of the feature attributions equals $F(x) = \sum_i A_i^F(x)$

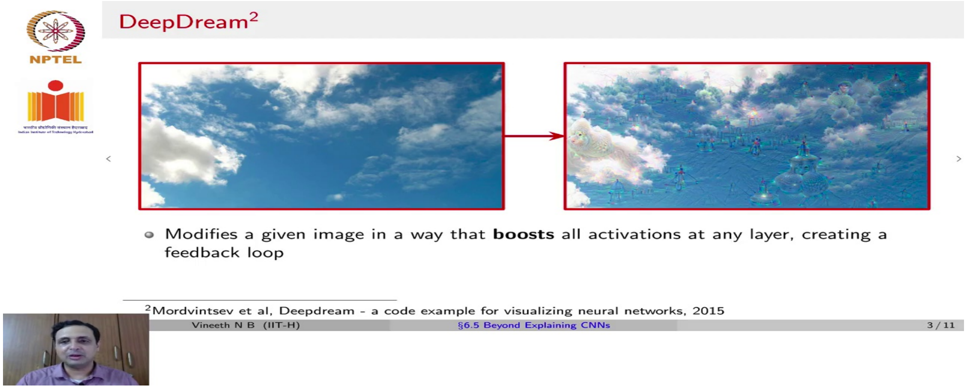
Sensitivity
If x has only one non-zero feature and $F(x) \neq 0$, then the attribution to that feature should be non-zero

Implementation Invariance
When two neural networks compute the same mathematical function $F(x)$, regardless of how differently they are implemented, the attributions to all features should always be identical.

Symmetry-Preserving
For any input x where the two values of two symmetric features are the same, their attributions should be identical as well.

[1] Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

Vineeth N B (IIT-H) §6.5 Beyond Explaining CNNs 2 / 11



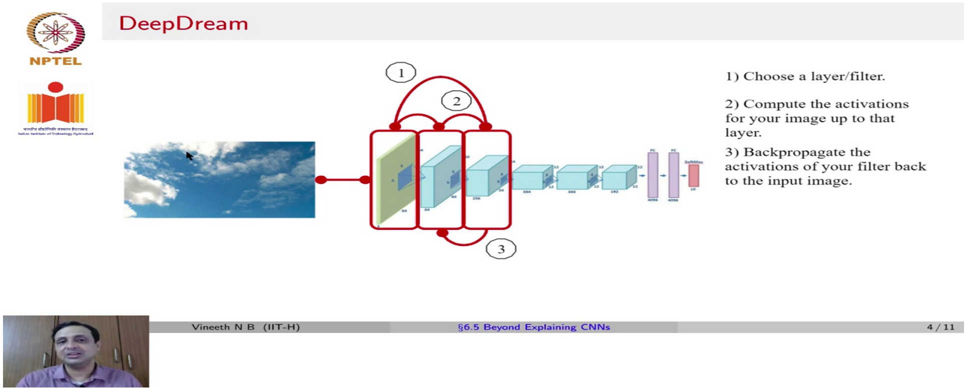
DeepDream[2]

- Modifies a given image in a way that **boosts** all activations at any layer, creating a feedback loop

[2] Mordvintsev et al, Deepdream - a code example for visualizing neural networks, 2015

Vineeth N B (IIT-H) §6.5 Beyond Explaining CNNs 3 / 11

But this time, the input is not a black image, or a gray image.This time, the input is just any other image that you want to juxtapose or overlay that input onto.



DeepDream

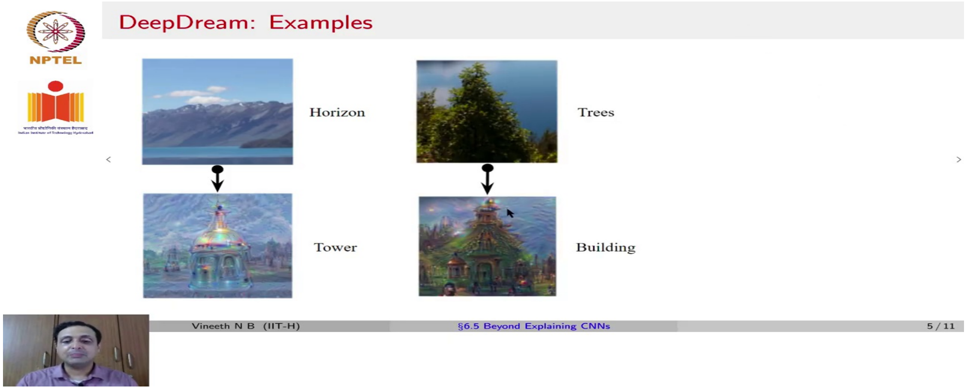
1) Choose a layer/filter.

2) Compute the activations for your image up to that layer.

3) Backpropagate the activations of your filter back to the input image.

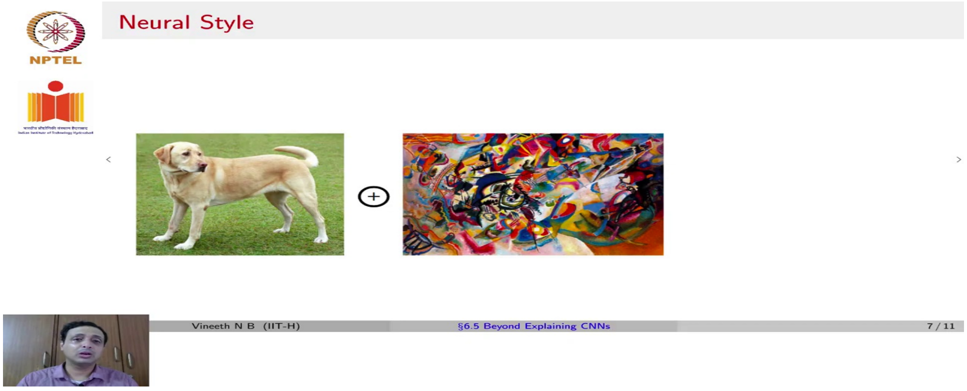
Vineeth N B (IIT-H) §6.5 Beyond Explaining CNNs 4 / 11

So you take the an image of the sky, take a face of the dog or any other filter, and keep adding to this image, the gradient that you get from the face of the dog, you try to see what in the input image would have resulted in that filter being fired, which would be given by the gradient of that filter the respect to the input.Remember, we talked about, if you wanted to maximize a filters activation, you said the gradient of only those filters to one, everything else to 0, and backprop to image and update the image using Gradient Ascent, you do the same thing here.What happens, you now see that you will get an image such as this, in this case, it is not a dog, it looks like a bunch of buildings that you add to the original image.And then you go back to 2 again, you input this image into the neural network, again, maybe you want to use the same filter, maybe you want to use a different filter.You could now take a different filter in a different layer.And now you could add that, and you know end up getting a different kind of a construction of the image.



DeepDream: Examples

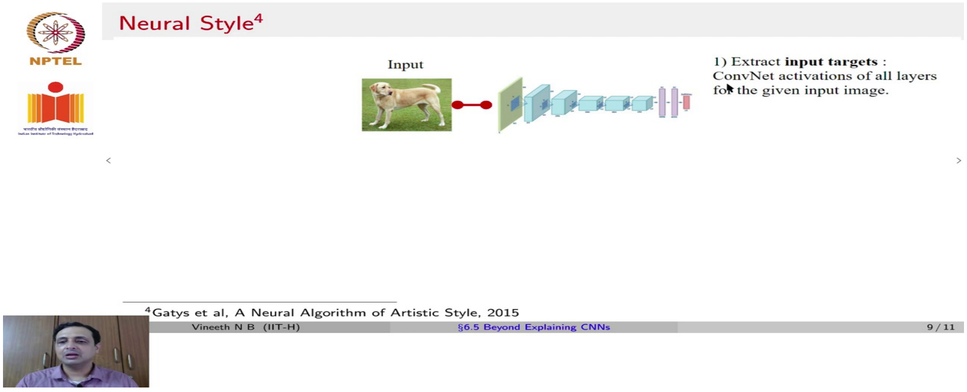
Vineeth N B (IIT-H) §6.5 Beyond Explaining CNNs 5 / 11



Neural Style

Vineeth N B (IIT-H) §6.5 Beyond Explaining CNNs 7 / 11

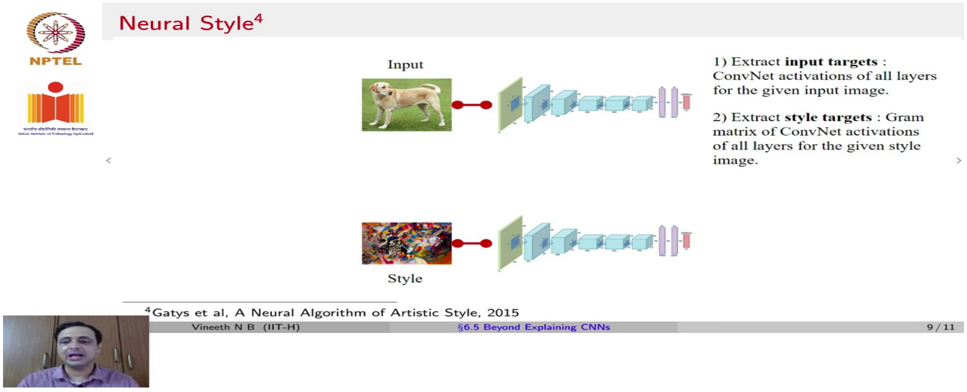
You want to know how do I get this image in this style? For example, can I take my photograph and make it look like Van Gogh had painted it? How would I get that style into my photo? That is what we want to study here.One is you could simply overlay take one of Van Gogh's paintings, overlay that on top of an image that I have, but unfortunately, this really does not give good results.Remember, in the scope of this entire week, we already have a pre trained model, we are not training the model in any way, we are only working with the train model in different ways.



Neural Style[4]

1) Extract **input targets** : ConvNet activations of all layers for the given input image.

[4] Gatys et al, A Neural Algorithm of Artistic Style, 2015



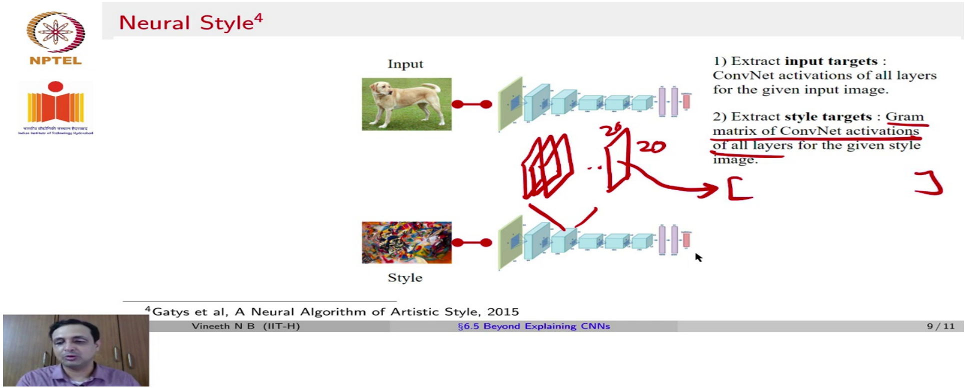
Neural Style[4]

1) Extract **input targets** : ConvNet activations of all layers for the given input image.

2) Extract **style targets** : Gram matrix of ConvNet activations of all layers for the given style image.

[4] Gatys et al, A Neural Algorithm of Artistic Style, 2015

And now, you take the covariance of this 400-dimensional vector with the another channel or another feature map in that volume.



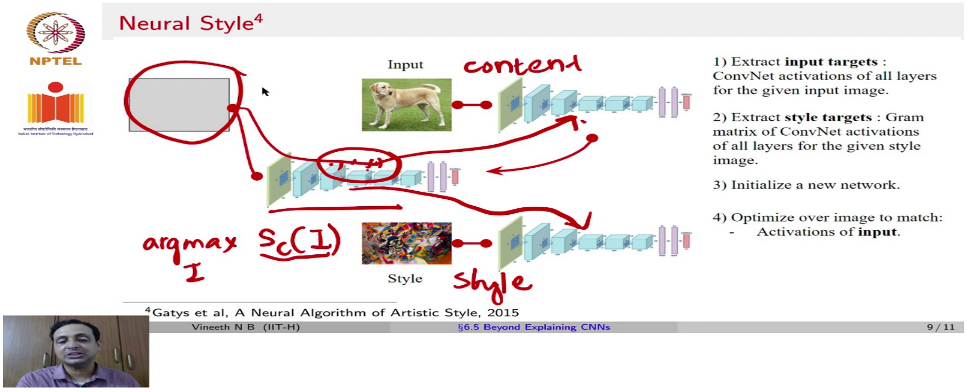
Neural Style[4]

1) Extract **input targets** : ConvNet activations of all layers for the given input image.

2) Extract **style targets** : Gram matrix of ConvNet activations of all layers for the given style image.

[4] Gatys et al, A Neural Algorithm of Artistic Style, 2015

And you would now get one such value of the covariance of one channel with the next channel.Similarly, you compute the covariance of every channel with every other channel, and you would get a Gram matrix of ConvNet activations of all layers.So, what do we do with the style target, once you obtain these Gram matrix of ConvNet activations from the style image, now, you take a new you again, take a network, a train model, and now you give a blank input as an image.And to this image, you backprop to image again, but you try to ensure that the activations that you get at any layer of this model is close to the activations that you get for this image on the same model in a different context.And the Gram matrix of activations that you get here resemble the gram matrix of activations that you get for the style model.So, you now backprop to image for a gray image, and another of these Alex Net models or any CNN model, you do backprop to image, remember that we said was argmax of I.But this time, we are going to try to minimize the distance between the activations that we get for this input here, and the activations that we get for this dog image on this network.



Neural Style[4]

$\underset{I}{\text{argmax}}\ S_c(I)$

1) Extract **input targets** : ConvNet activations of all layers for the given input image.

2) Extract **style targets** : Gram matrix of ConvNet activations of all layers for the given style image.

3) Initialize a new network.

4) Optimize over image to match:
- Activations of **input**.

[4] Gatys et al, A Neural Algorithm of Artistic Style, 2015

Vineeth N B (IIT-H) §6.5 Beyond Explaining CNNs 9 / 11

when you combine these two, you end up getting an image, which takes the style from the style image and the content from the input image.So you can see here is the input image, and here is the style and you get a pretty interesting output that resembles the style.More examples here, if you would like to take a look at where the content and style is varied, and you have an entire grid of several examples of taking a content and trying different styles on them.So, if you would like to understand this more, there are a couple of interesting links here, which gives you an intuitive understanding of neural style transfer.