Neural Style Transfer

A presentation for Artificial intelligence class

Yara Mohamadi Bahram

9430833

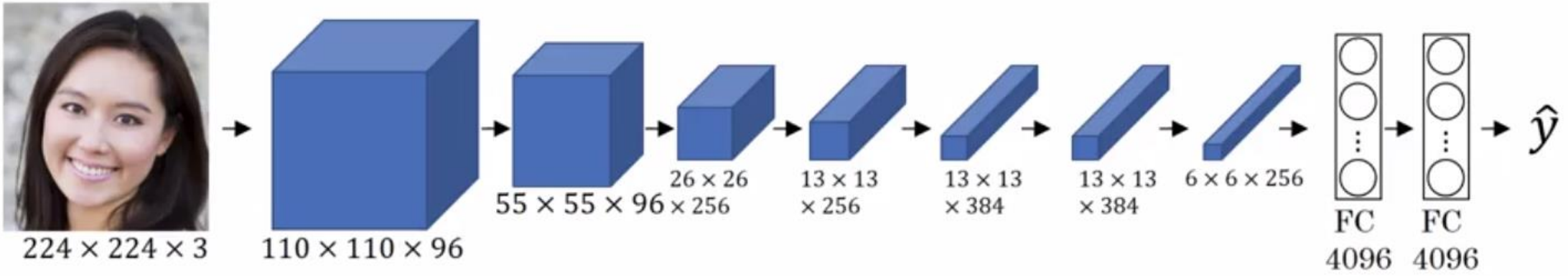
What is Neural Style Transfer

Content image (C)

Style image (S)

Generated image (G)

What are Deep Convolutional Networks learning?



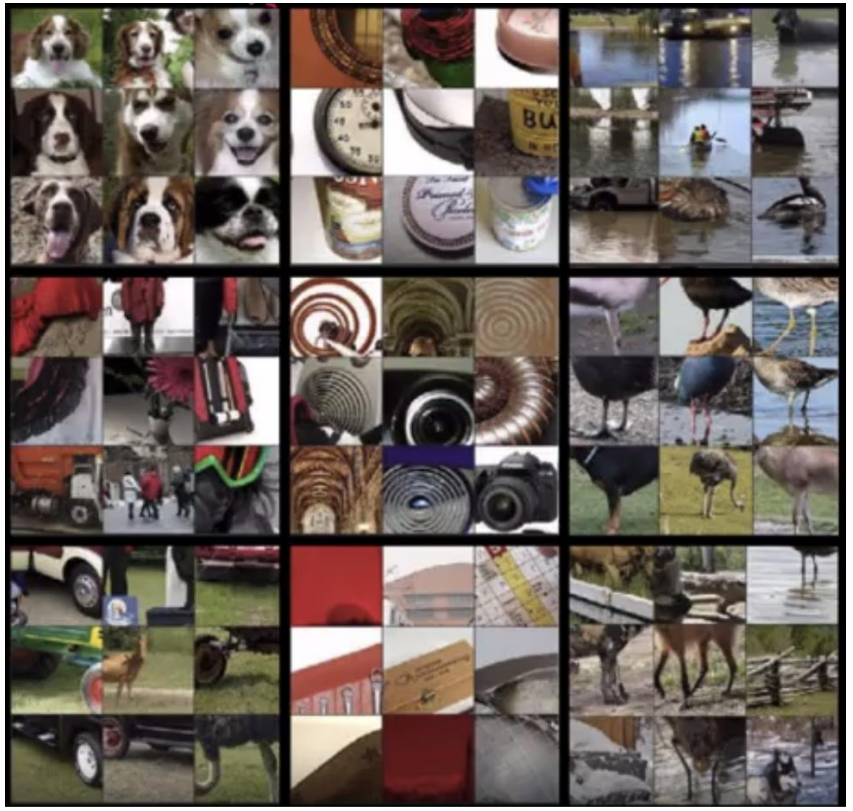
What happens in each layer?

(0,0) (0,1) (0,2)

(0,0)

(1,0)

(2,0)



9 Image patches that activate random hidden units in layer[n] the most

- ✓ Layer 1: Simple patterns like edges or colors
- ✓ Layer 2: a little more complex like textures, or round patterns
- ✓ Layer 4: Object classifications

Finding the generated image G

1. Initialize G randomly

   G: 100 x 100 x 3

2. Use gradient descent to minimize J(G) G = G – d/dG(J(G))

$$J(G) = \alpha\, J_{content}(C, G) + \beta\, J_{style}\,(S, G)$$

Content cost function

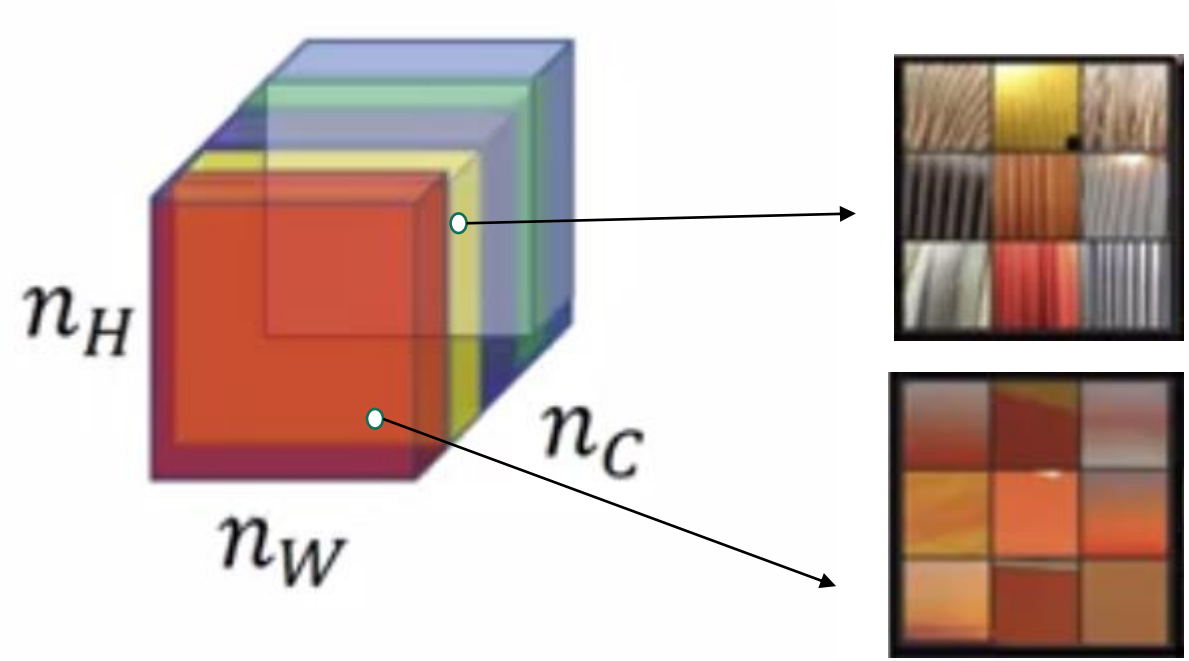
- We choose hidden layer L to compute our cost function
  - Too small: pixel wise similarity
  - Too big: E.g. if there is a dog in my content image, I want a dog in my generated image.
- Use pre-trained ConvNets. (E.g. VGG)
- a[L] : activation of layer L on the images
- If a[L](C) and a[L](G) are similar, then they are similar

$$J_{content}(C,G) = \frac{1}{2} ||a[L](C) - a[L](G)||^2$$

Style cost function

Lets measure style for layer L

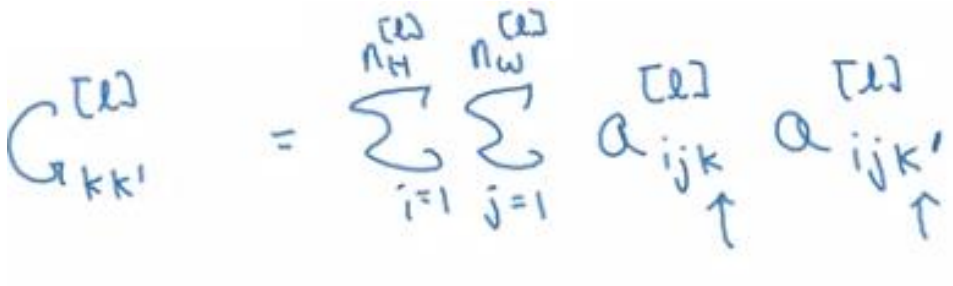
- How correlated are the activations across different channels?



Correlated vs. Uncorrelated

- How often they occur or don't occur together?
- Results in a matrix containing style of an image!

Style matrix



J_style[L](S, G) = || G[L](s) – G[L](G) ||^2

Includes normalization parameter

J_style(S, G) = sum of J_style[L](S, G) for all layers

A review on our cost function

$$J(G) = \alpha\, J_{content}(C, G) + \beta\, J_{style}\,(S, G)$$