



deeplearning.ai

Neural Style Transfer

What is neural style
transfer?

Neural style transfer



Content (c)



Style (s)



Generated image ($c \cdot s$)



Content (c)



Style (s)



Generated image ($c \cdot s$)

[Images generated by Justin Johnson]

Andrew Ng

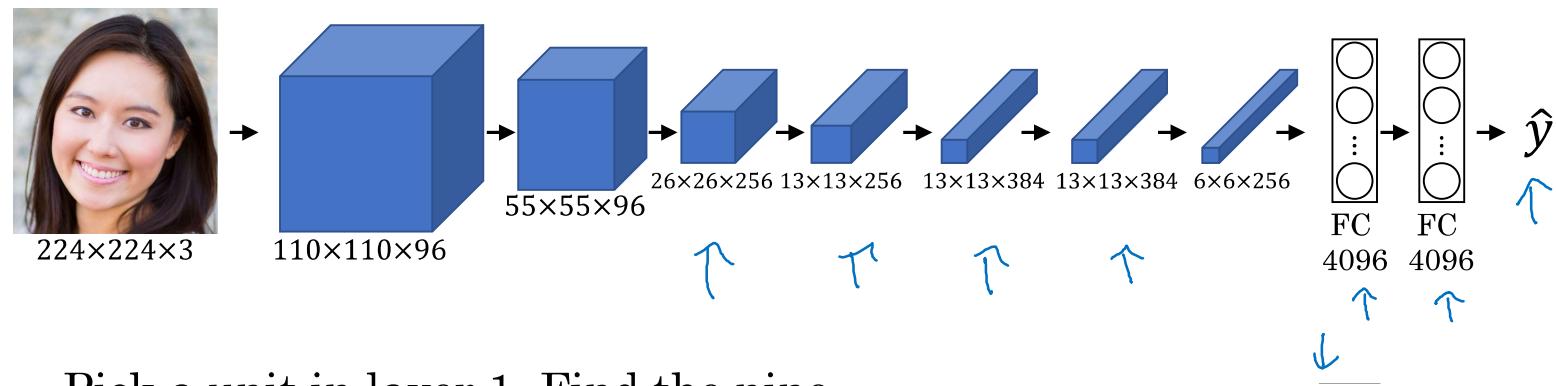


deeplearning.ai

Neural Style Transfer

What are deep
ConvNets learning?

Visualizing what a deep network is learning



Pick a unit in layer 1. Find the nine image patches that maximize the unit's activation.

Repeat for other units.



[Zeiler and Fergus., 2013, Visualizing and understanding convolutional networks]

Andrew Ng

Visualizing deep layers



Layer 1



Layer 2



Layer 3



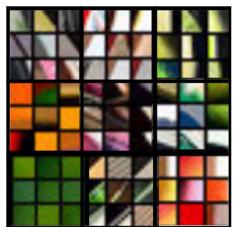
Layer 4



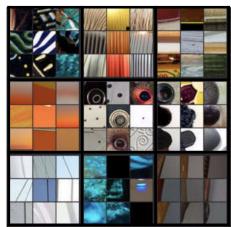
Layer 5

Andrew Ng

Visualizing deep layers: Layer 1



Layer 1



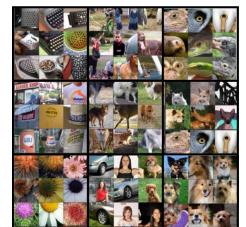
Layer 2



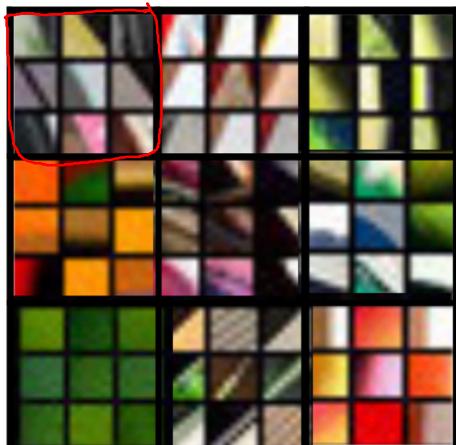
Layer 3



Layer 4

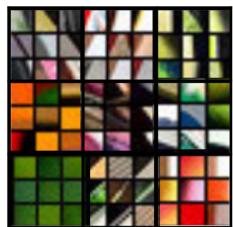


Layer 5

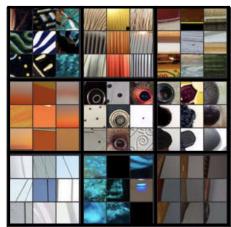


Andrew Ng

Visualizing deep layers: Layer 2



Layer 1



Layer 2



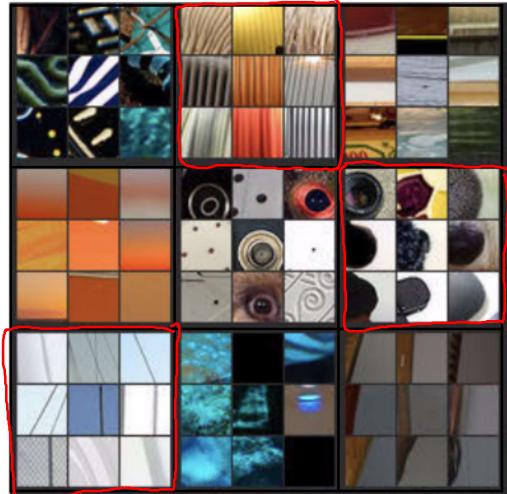
Layer 3



Layer 4



Layer 5

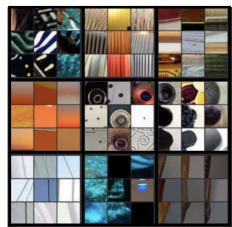


Andrew Ng

Visualizing deep layers: Layer 3



Layer 1



Layer 2



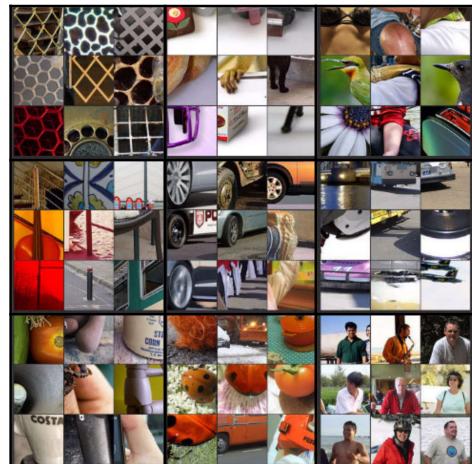
Layer 3



Layer 4

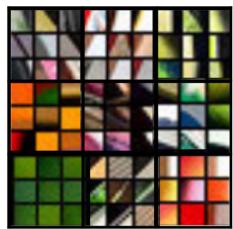


Layer 5

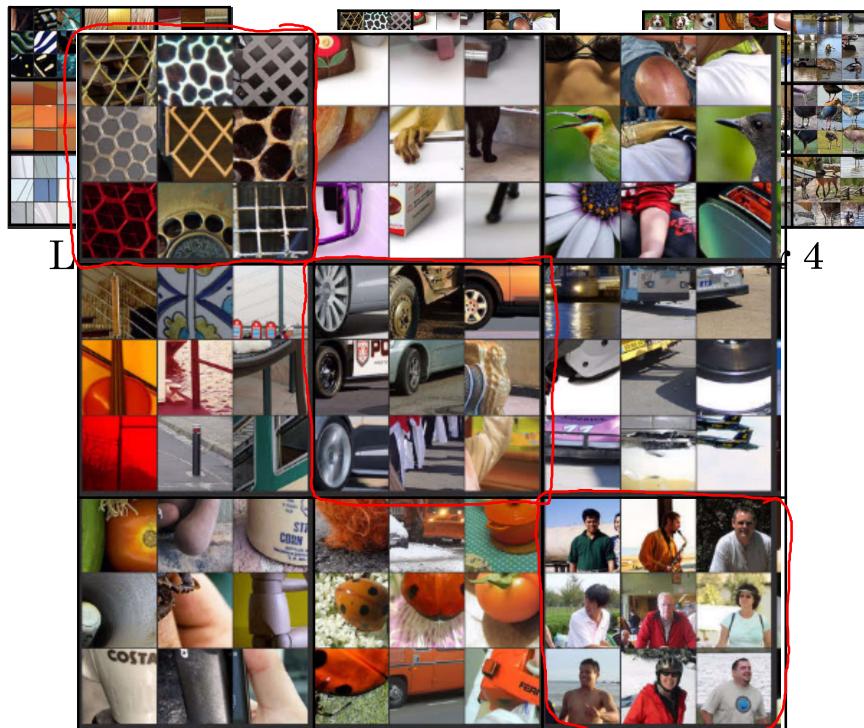


Andrew Ng

Visualizing deep layers: Layer 3



Layer 1



L



Layer 5

Andrew Ng

Visualizing deep layers: Layer 4

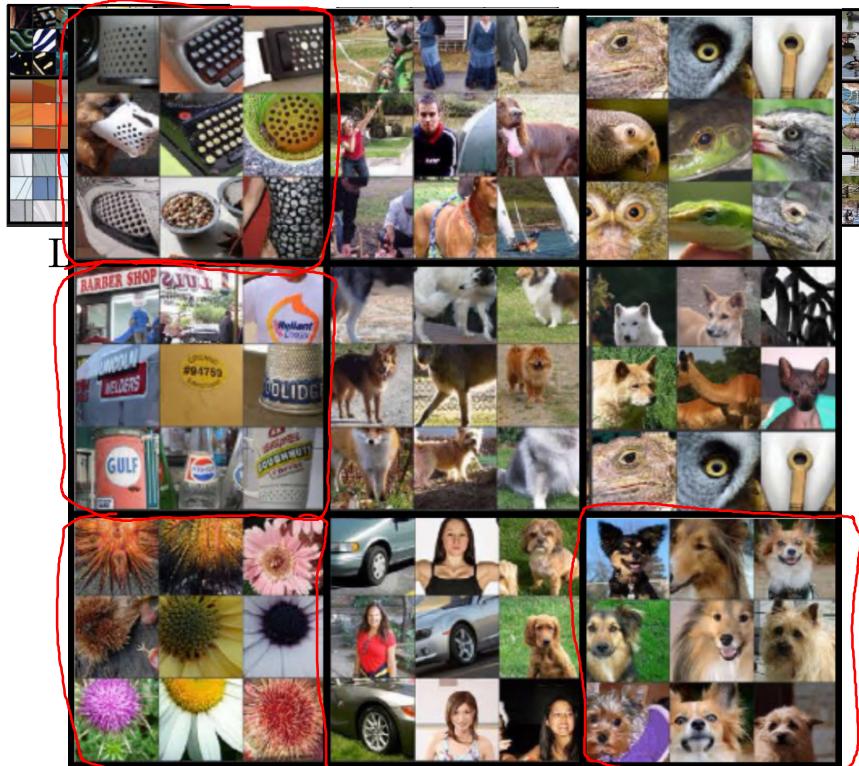


Andrew Ng

Visualizing deep layers: Layer 5



Layer 1



Layer 5

Andrew Ng

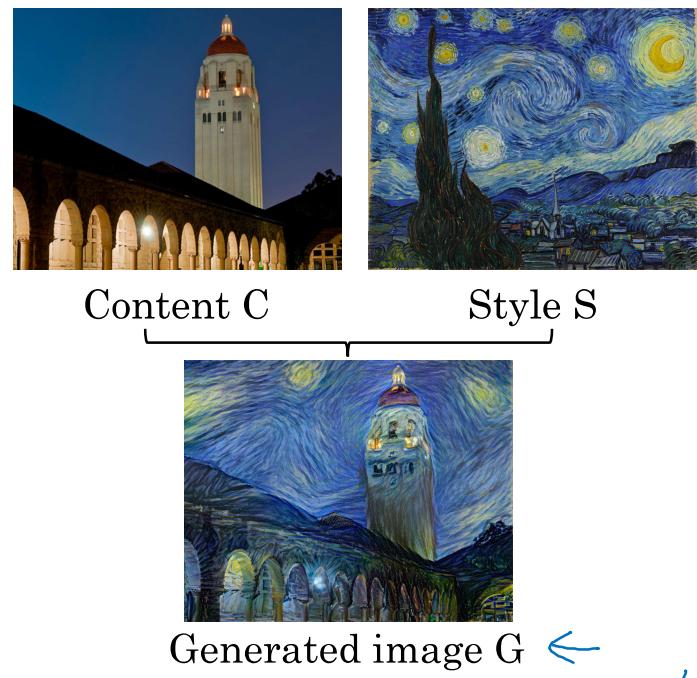


deeplearning.ai

Neural Style Transfer

Cost function

Neural style transfer cost function



$$\begin{aligned} J(G) = & \alpha J_{\text{Content}}(C, G) \\ & + \beta J_{\text{Style}}(S, G) \end{aligned}$$

[Gatys et al., 2015. A neural algorithm of artistic style. Images on slide generated by Justin Johnson] Andrew Ng

Find the generated image G

1. Initiate G randomly

$G: 100 \times 100 \times 3$

$\xrightarrow{\text{RGB}}$



2. Use gradient descent to minimize $J(G)$

$$G := G - \frac{\partial}{\partial G} J(G)$$



[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng



deeplearning.ai

Neural Style Transfer

Content cost function

Content cost function

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

- Say you use hidden layer \underline{l} to compute content cost.
- Use pre-trained ConvNet. (E.g., VGG network)
- Let $\underline{a^{[l](C)}}$ and $\underline{a^{[l](G)}}$ be the activation of layer \underline{l} on the images
- If $a^{[l](C)}$ and $a^{[l](G)}$ are similar, both images have similar content

$$J_{content}(C, G) = \frac{1}{2} \| \underbrace{a^{[l](C)}}_{\text{activation}} - \underbrace{a^{[l](G)}}_{\text{activation}} \|_2^2$$

[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

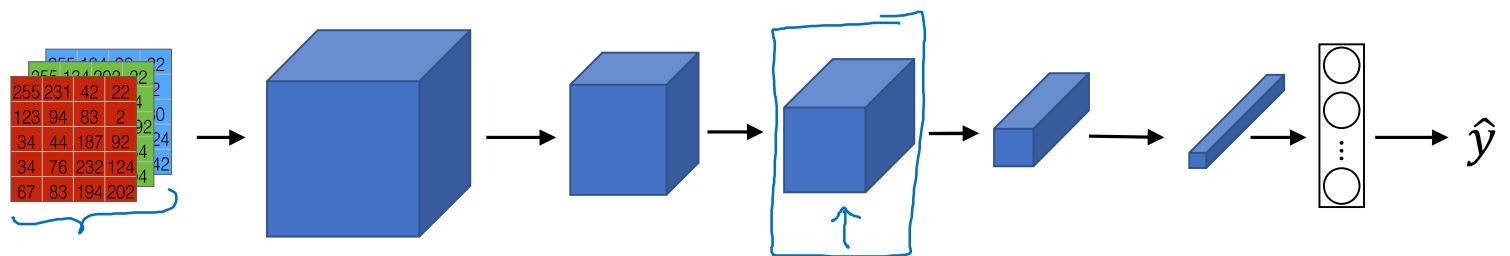


deeplearning.ai

Neural Style Transfer

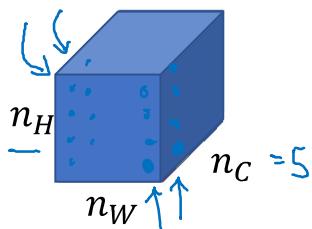
Style cost function

Meaning of the “style” of an image



Say you are using layer l 's activation to measure “style.”

Define style as correlation between activations across channels.



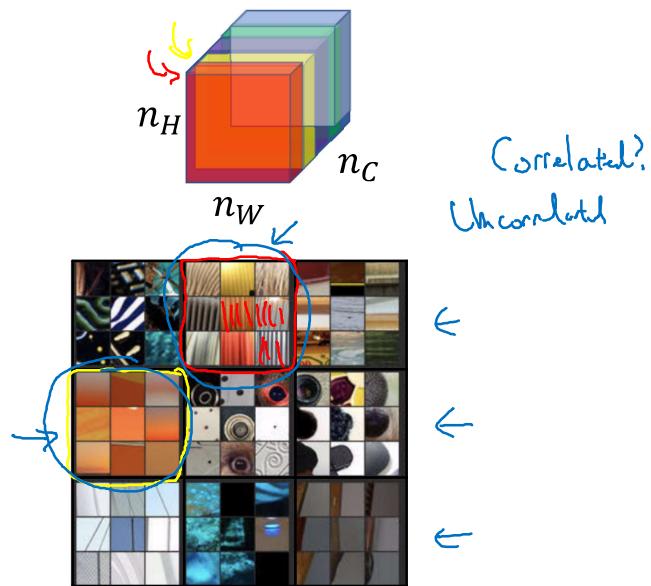
How correlated are the activations across different channels?

[Gatys et al., 2015. A neural algorithm of artistic style]

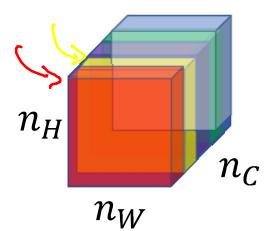
Andrew Ng

Intuition about style of an image

Style image



Generated Image



[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

Style matrix

Let $a_{i,j,k}^{[l]}$ = activation at (i, j, k) . $G^{[l]}$ is $n_c^{[l]} \times n_c^{[l]}$

$$\rightarrow G_{kk'}^{[l](s)} = \sum_{i=1}^{n_h^{[l]}} \sum_{j=1}^{n_w^{[l]}} a_{ijk}^{[l](s)} a_{ijk'}^{[l](s)}$$

$$\rightarrow G_{kk'}^{[l](G)} = \sum_{i=1}^{n_h^{[l]}} \sum_{j=1}^{n_w^{[l]}} a_{ijk}^{[l](G)} a_{ijk'}^{[l](G)}$$

H
W
C

n_c
 $G_{kk'}^{[l]}$
 $k, k' = 1, \dots, n_c^{[l]}$

"Gram matrix"

$$J_{\text{style}}^{[l]}(S, G) = \frac{1}{\beta} \| G^{[l](s)} - G^{[l](G)} \|_F^2$$

$$= \frac{1}{(2n_h^{[l]} n_w^{[l]} n_c^{[l]})^2} \sum_k \sum_{k'} (G_{kk'}^{[l](s)} - G_{kk'}^{[l](G)})^2$$

Style cost function

$$\left\| G^{[l](S)} - G^{[l](G)} \right\|_F^2$$

$$J_{style}^{[l]}(S, G) = \frac{1}{\left(2n_H^{[l]} n_W^{[l]} n_C^{[l]}\right)^2} \sum_k \sum_{k'} (G_{kk'}^{[l](S)} - G_{kk'}^{[l](G)})$$

$$J_{style}(S, G) = \sum_l J_{style}^{[l]}(S, G)$$

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

[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

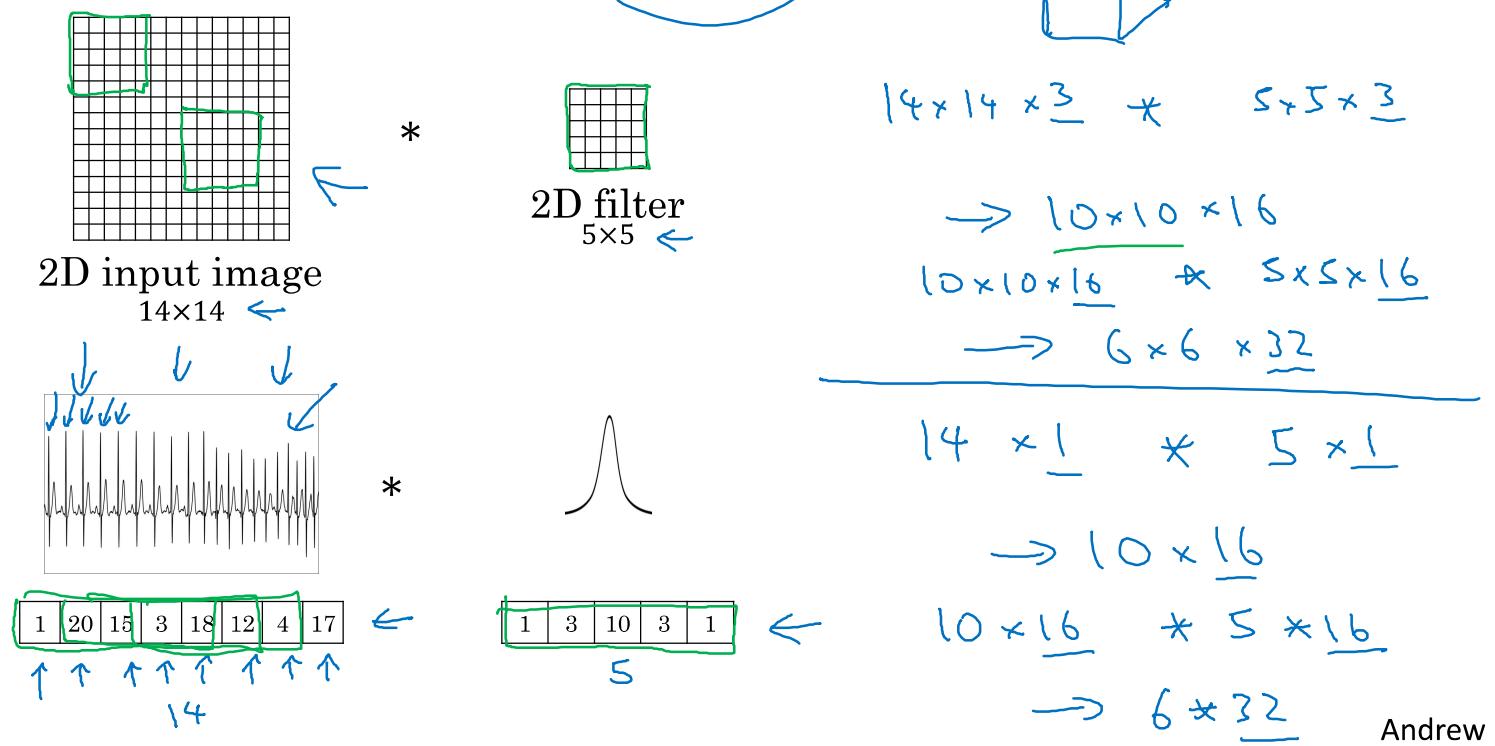


deeplearning.ai

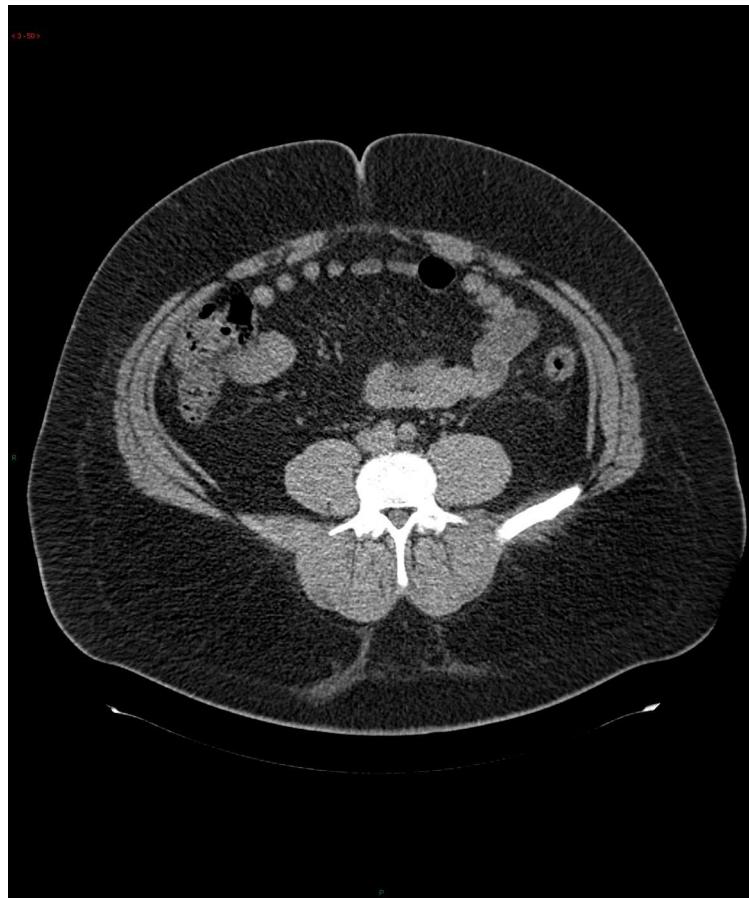
Convolutional Networks in 1D or 3D

1D and 3D
generalizations of
models

Convolutions in 2D and 1D



3D data



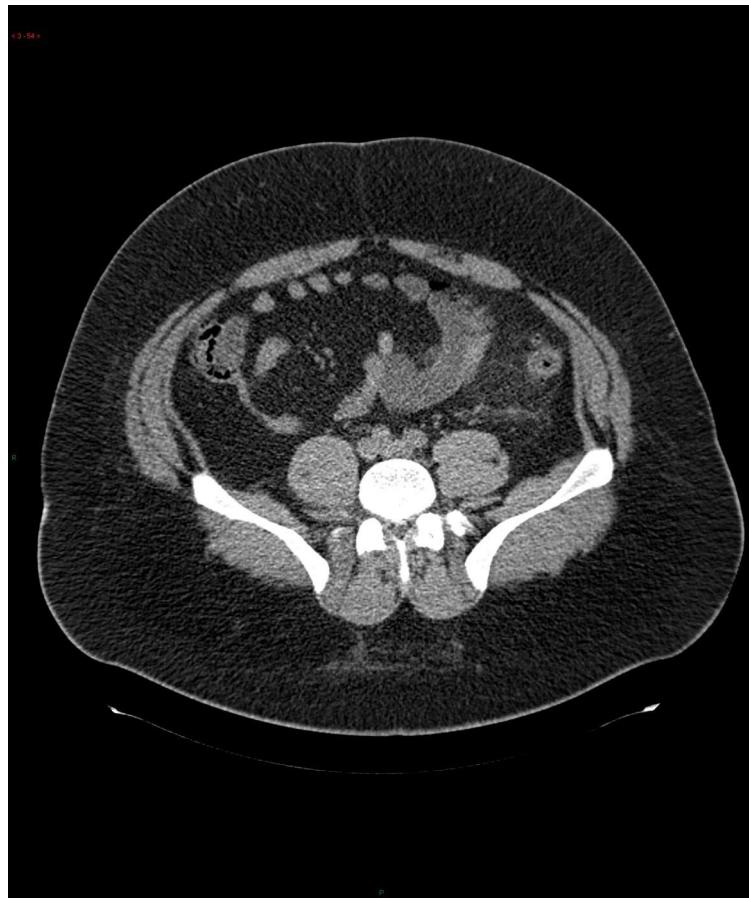
Andrew Ng

3D data



Andrew Ng

3D data



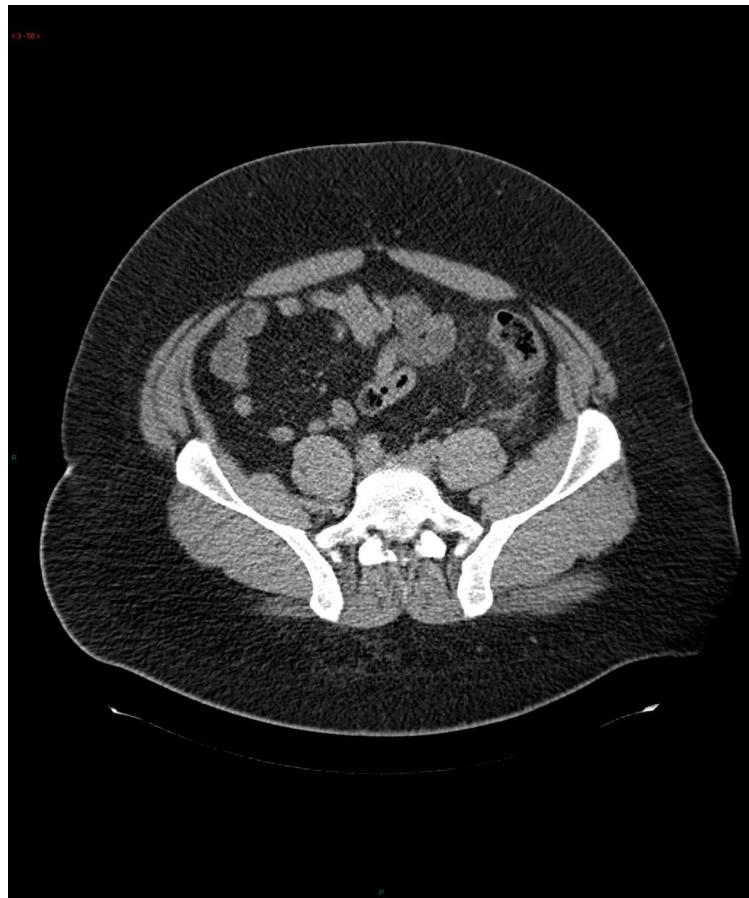
Andrew Ng

3D data



Andrew Ng

3D data



Andrew Ng

3D data



Andrew Ng

3D data



Andrew Ng

3D data



Andrew Ng

3D data



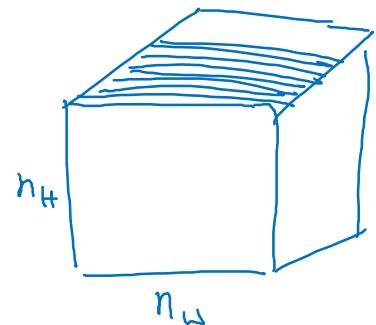
Andrew Ng

3D data



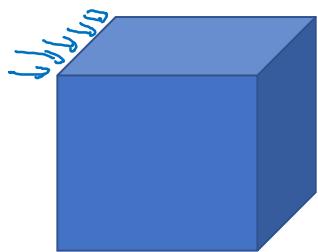
Andrew Ng

3D data

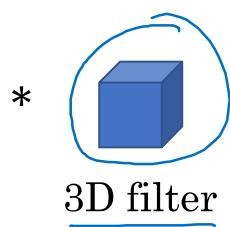


Andrew Ng

3D convolution



3D volume



$$\begin{array}{c} \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \underbrace{14 \times 14 \times 14}_{\text{Input}} \times 1 \\ * \underbrace{5 \times 5 \times 5 \times 1}_{\text{16 filters.}} \\ \rightarrow 10 \times 10 \times 10 \times 16 \\ * \underbrace{5 \times 5 \times 5 \times 16}_{\text{32 filters.}} \\ \rightarrow 6 \times 6 \times 6 \times 32 \end{array}$$

Andrew Ng