

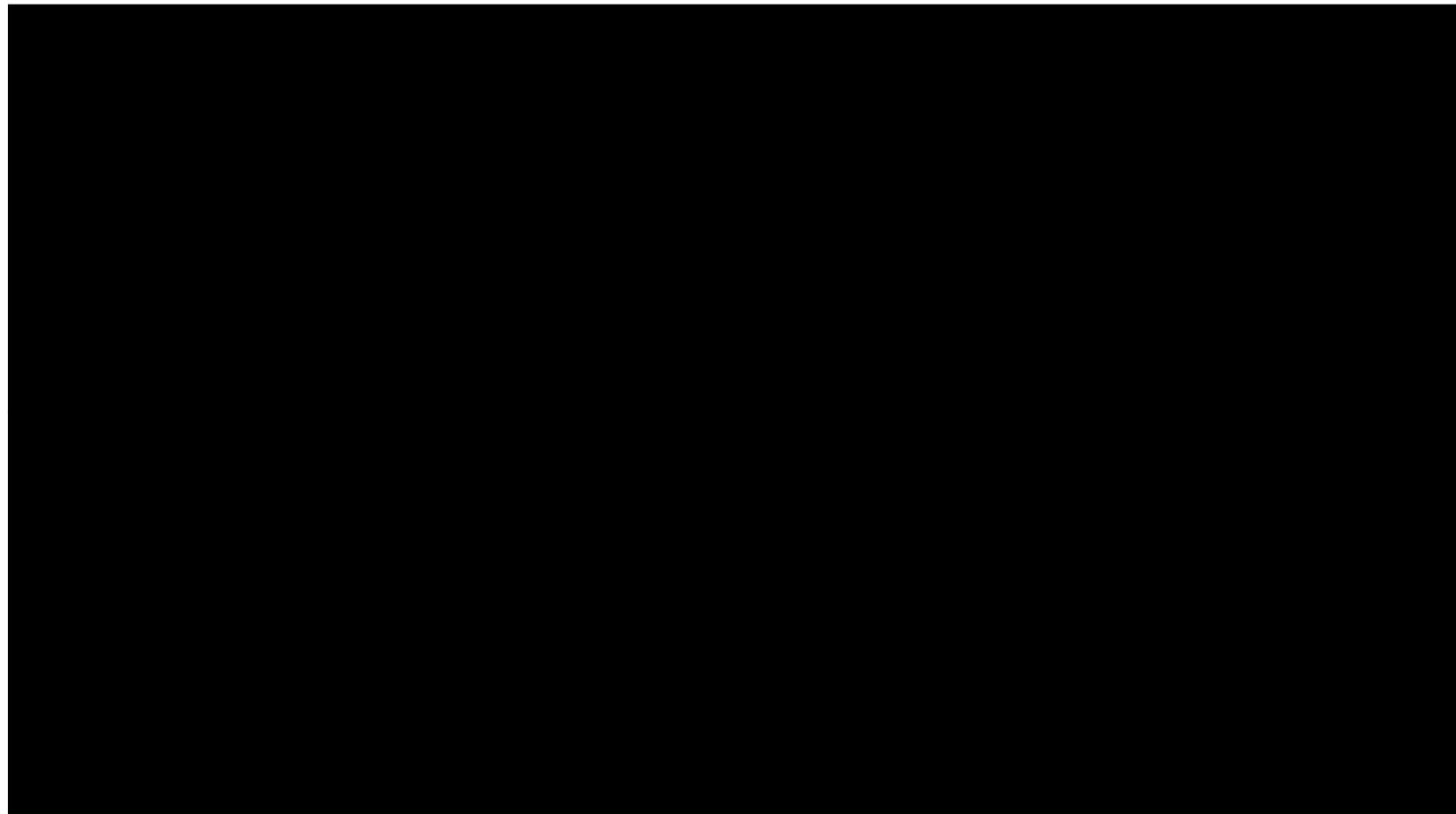


deeplearning.ai

Face recognition

What is face
recognition?

Face recognition



[Courtesy of Baidu]

Andrew Ng

Face verification vs. face recognition

→ Verification

- Input image, name/ID
- Output whether the input image is that of the claimed person

1:1

99%

99.9
~~~

## → Recognition

- Has a database of K persons
- Get an input image
- Output ID if the image is any of the K persons (or “not recognized”)

1:K

K=100 ←



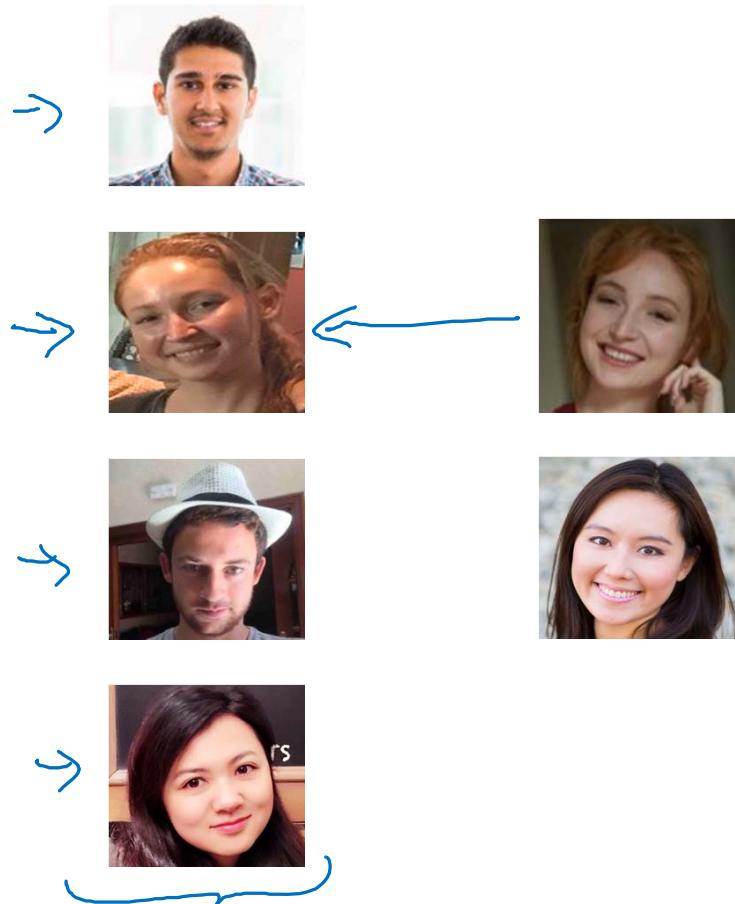
deeplearning.ai

# Face recognition

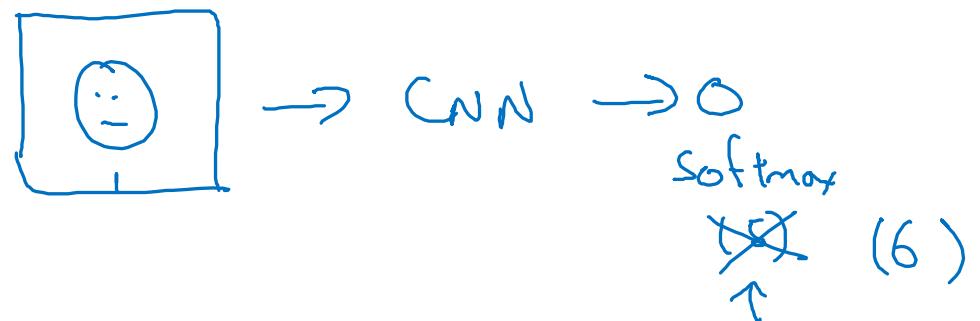
---

# One-shot learning

# One-shot learning



Learning from one  
example to recognize the  
person again



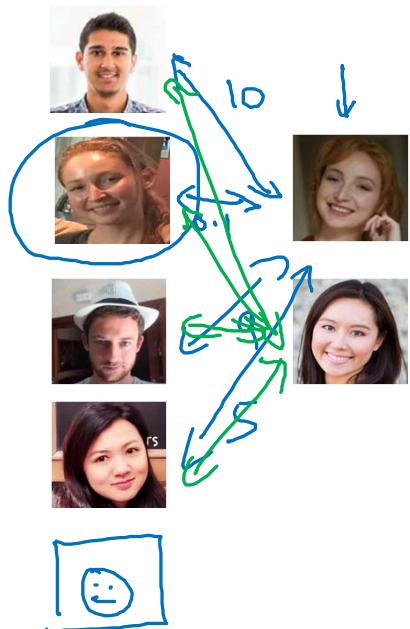
Andrew Ng

# Learning a “similarity” function

→  $d(\underline{\text{img1}}, \underline{\text{img2}}) = \text{degree of difference between images}$

If  $d(\text{img1}, \text{img2}) \leq \tau$       "same"  
 $> \tau$       "different"

} Verification.



Andrew Ng



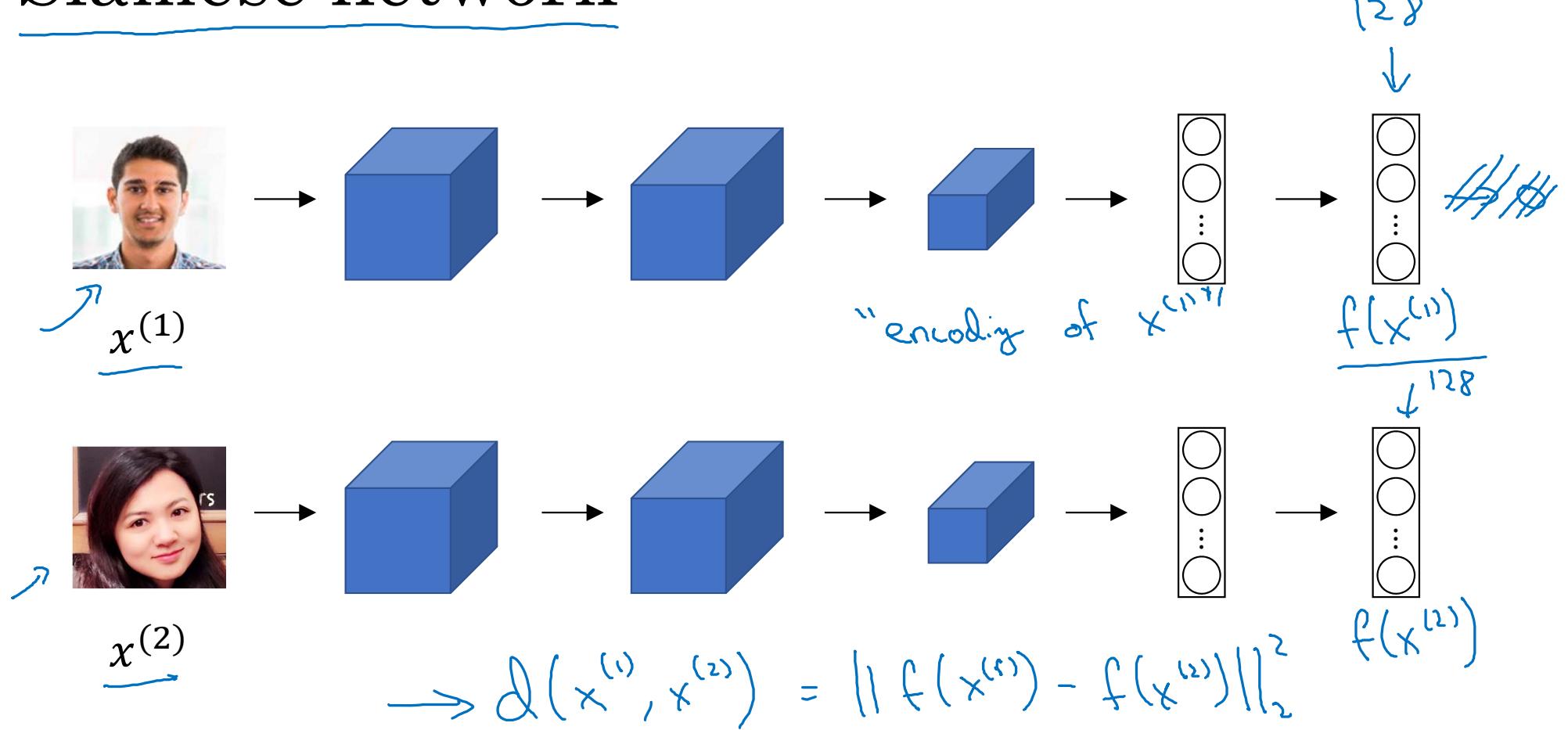
deeplearning.ai

# Face recognition

---

## Siamese network

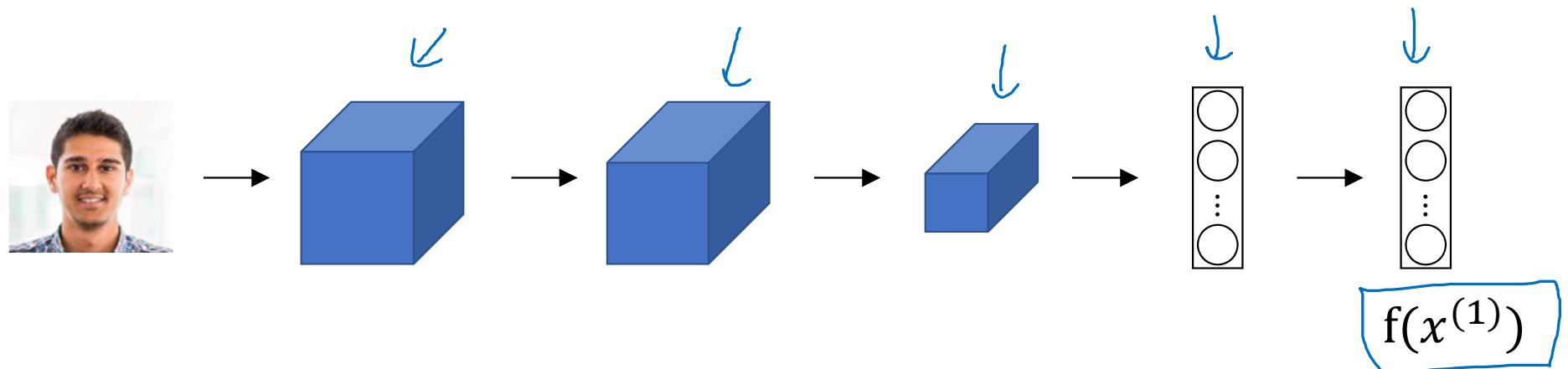
# Siamese network



[Taigman et. al., 2014. DeepFace closing the gap to human level performance]

Andrew Ng

# Goal of learning



Parameters of NN define an encoding  $f(x^{(i)})$  128

Learn parameters so that:

If  $x^{(i)}, x^{(j)}$  are the same person,  $\|f(x^{(i)}) - f(x^{(j)})\|^2$  is small.

If  $x^{(i)}, x^{(j)}$  are different persons,  $\|f(x^{(i)}) - f(x^{(j)})\|^2$  is large.

Andrew Ng



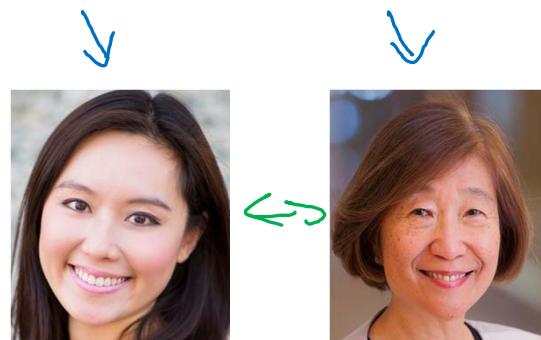
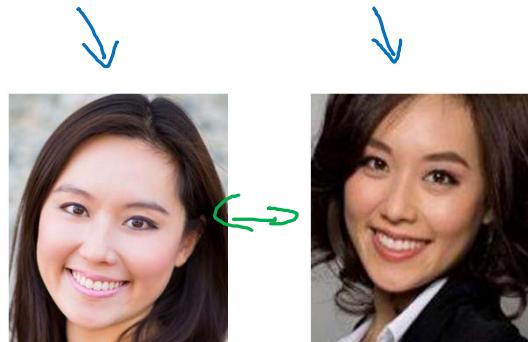
deeplearning.ai

# Face recognition

---

## Triplet loss

# Learning Objective



Anchor      Positive

$$A \quad d(A, P) = 0.5$$

Want:

$$\frac{\|f(A) - f(P)\|^2}{d(A, P)} + \alpha \leq 0.2$$

Anchor      Negative

$$A \quad d(A, N) = 0.5$$

$$\frac{\|f(A) - f(N)\|^2}{d(A, N)}$$

$$\frac{\|f(A) - f(P)\|^2}{\alpha} - \frac{\|f(A) - f(N)\|^2}{\alpha} + \alpha \leq 0 \quad \text{margin}$$

$f(\text{img}) = \vec{0}$

[Schroff et al., 2015, FaceNet: A unified embedding for face recognition and clustering]

Andrew Ng

# Loss function

Given 3 imgs

$A, P, N$ :

$$L(A, P, N) = \max \left( \frac{\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \lambda}{\epsilon \rho} > 0, 0 \right)$$

$$J = \sum_{i=1}^m L(A^{(i)}, P^{(i)}, N^{(i)})$$

$A, P$   
 $T \uparrow$

Training set:  $\underbrace{10k}_{\infty}$  pictures of  $\overline{1k}$  persons

# Choosing the triplets A,P,N



During training, if A,P,N are chosen randomly,  
 $d(A, P) + \alpha \leq d(A, N)$  is easily satisfied.

$$\underbrace{\|f(A) - f(P)\|^2}_{\|f(A) - f(N)\|^2} + \alpha \leq \underbrace{\|f(A) - f(N)\|^2}$$

Choose triplets that're “hard” to train on.

$$\begin{aligned} \cancel{d(A, P)} + \alpha &\leq d(A, N) \\ \frac{d(A, P)}{\downarrow} &\approx \frac{d(A, N)}{\uparrow} \end{aligned}$$

Face Net  
Deep Face



[Schroff et al., 2015, FaceNet: A unified embedding for face recognition and clustering]

Andrew Ng

# Training set using triplet loss

Anchor



Positive



Negative



:

:

:



J

$$d(x^{(i)}, x^{(j)})$$

Andrew Ng



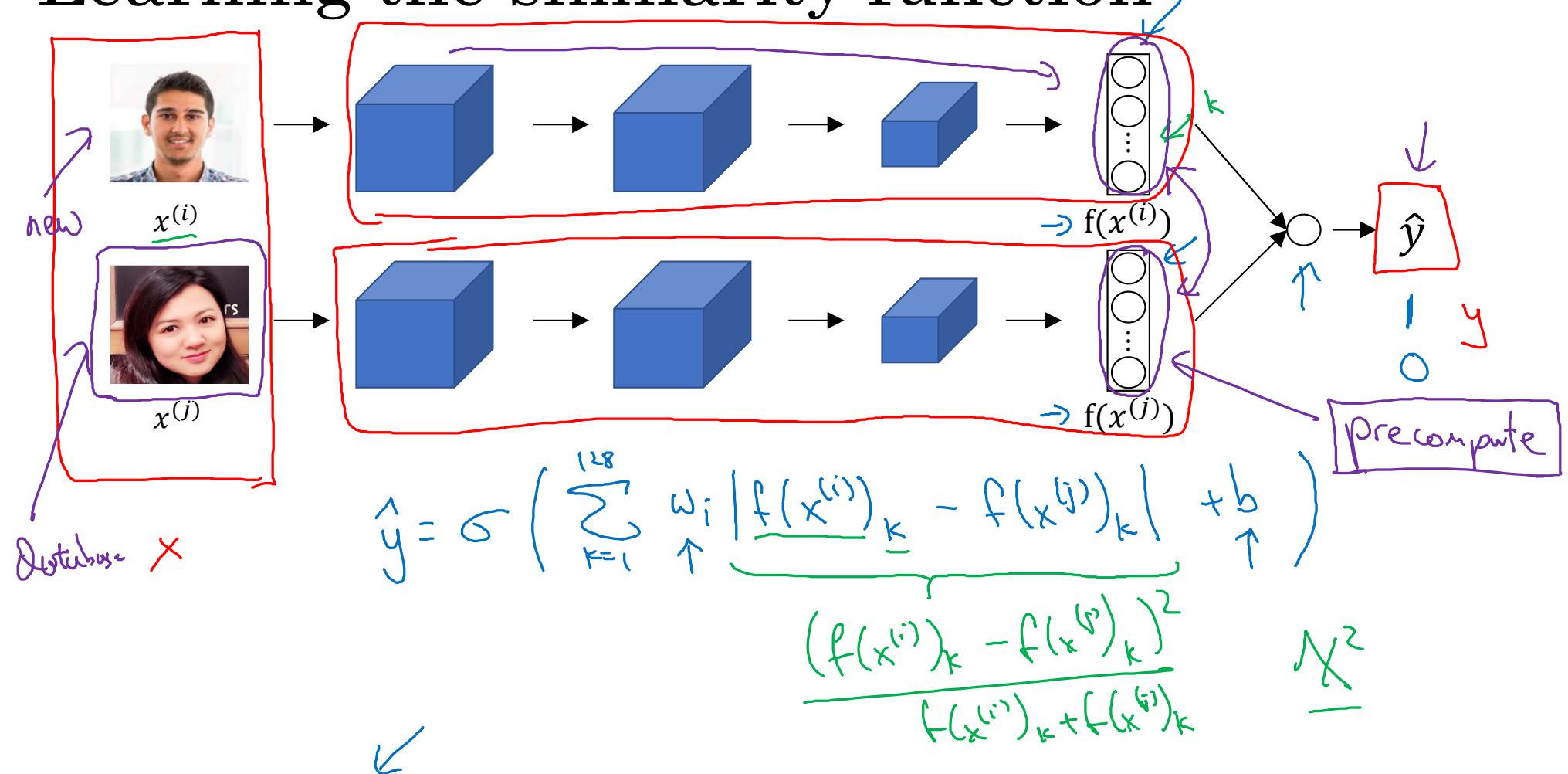
deeplearning.ai

## Face recognition

---

## Face verification and binary classification

# Learning the similarity function



[Taigman et. al., 2014. DeepFace closing the gap to human level performance]

Andrew Ng

# Face verification supervised learning

| $x$                                                                                 | $y$                                                                                                          |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
|    | <br>1<br><i>"Same"</i>      |
|    | <br>0<br><i>"Different"</i> |
|   | <br>0                      |
|  | <br>1                     |

[Taigman et. al., 2014. DeepFace closing the gap to human level performance]

Andrew Ng



deeplearning.ai

# Neural Style Transfer

---

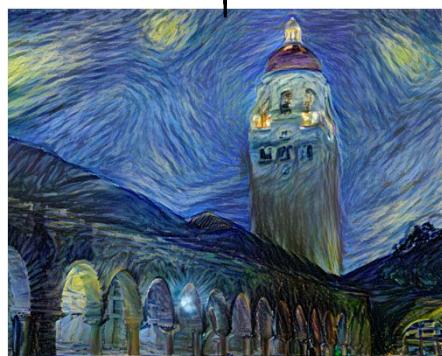
## What is neural style transfer?

# Neural style transfer



Content ( $c$ )

Style ( $s$ )



Generated image ( $g$ )



Content ( $c$ )

Style ( $s$ )



Generated image ( $g$ )

[Images generated by Justin Johnson]

Andrew Ng



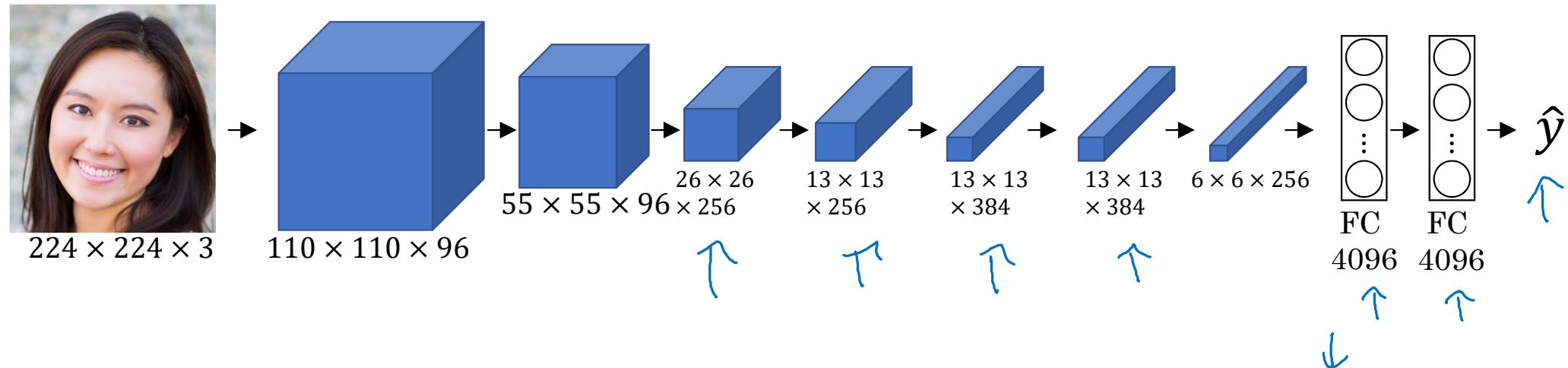
deeplearning.ai

# Neural Style Transfer

---

## What are deep ConvNets learning?

# Visualizing what a deep network is learning



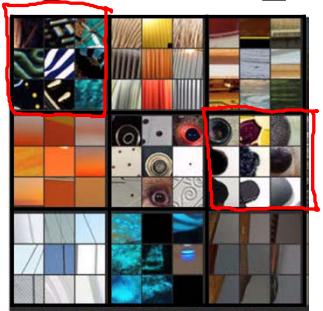
Pick a unit in layer 1. Find the nine image patches that maximize the unit's activation.

Repeat for other units.

# Visualizing deep layers



Layer 1



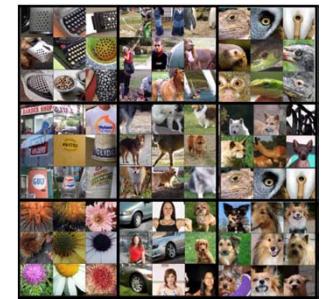
Layer 2



Layer 3



Layer 4



Layer 5

Andrew Ng

# Visualizing deep layers: Layer 1



Layer 1



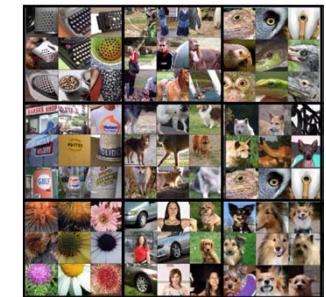
Layer 2



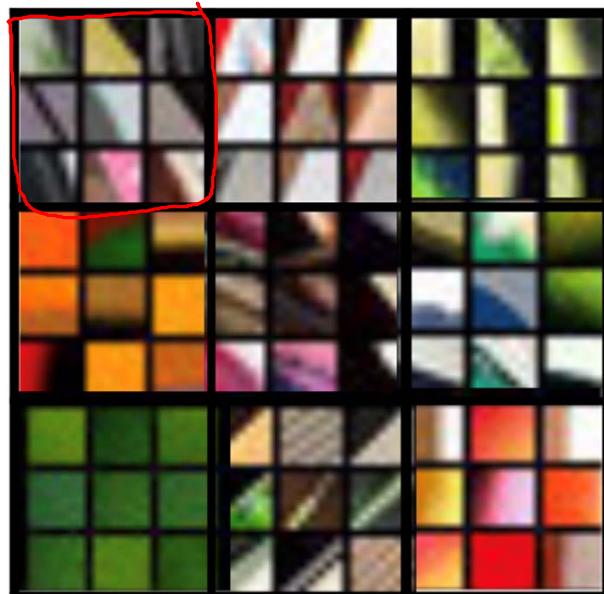
Layer 3



Layer 4



Layer 5



Andrew Ng

# Visualizing deep layers: Layer 2



Layer 1



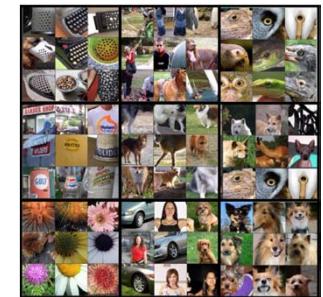
Layer 2



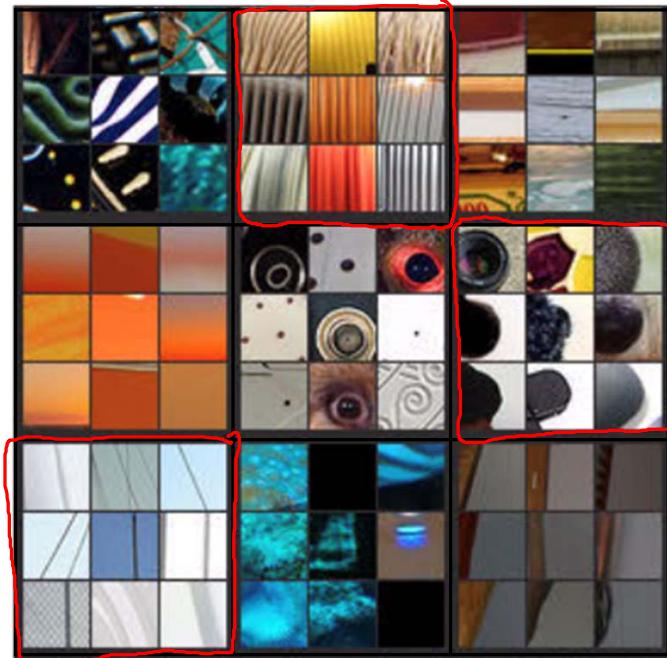
Layer 3



Layer 4



Layer 5



Andrew Ng

# Visualizing deep layers: Layer 3



Layer 1



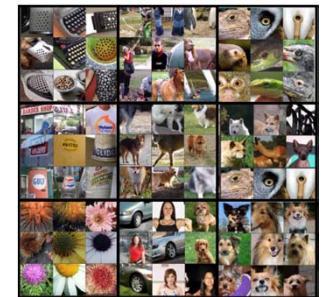
Layer 2



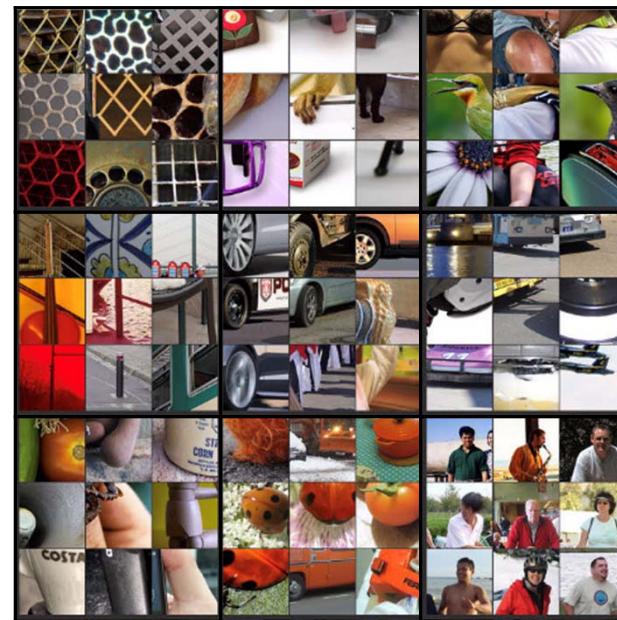
Layer 3



Layer 4



Layer 5

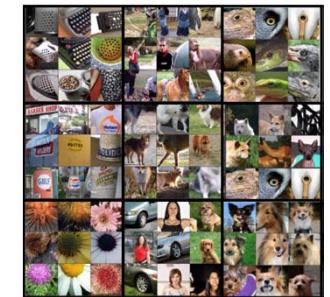
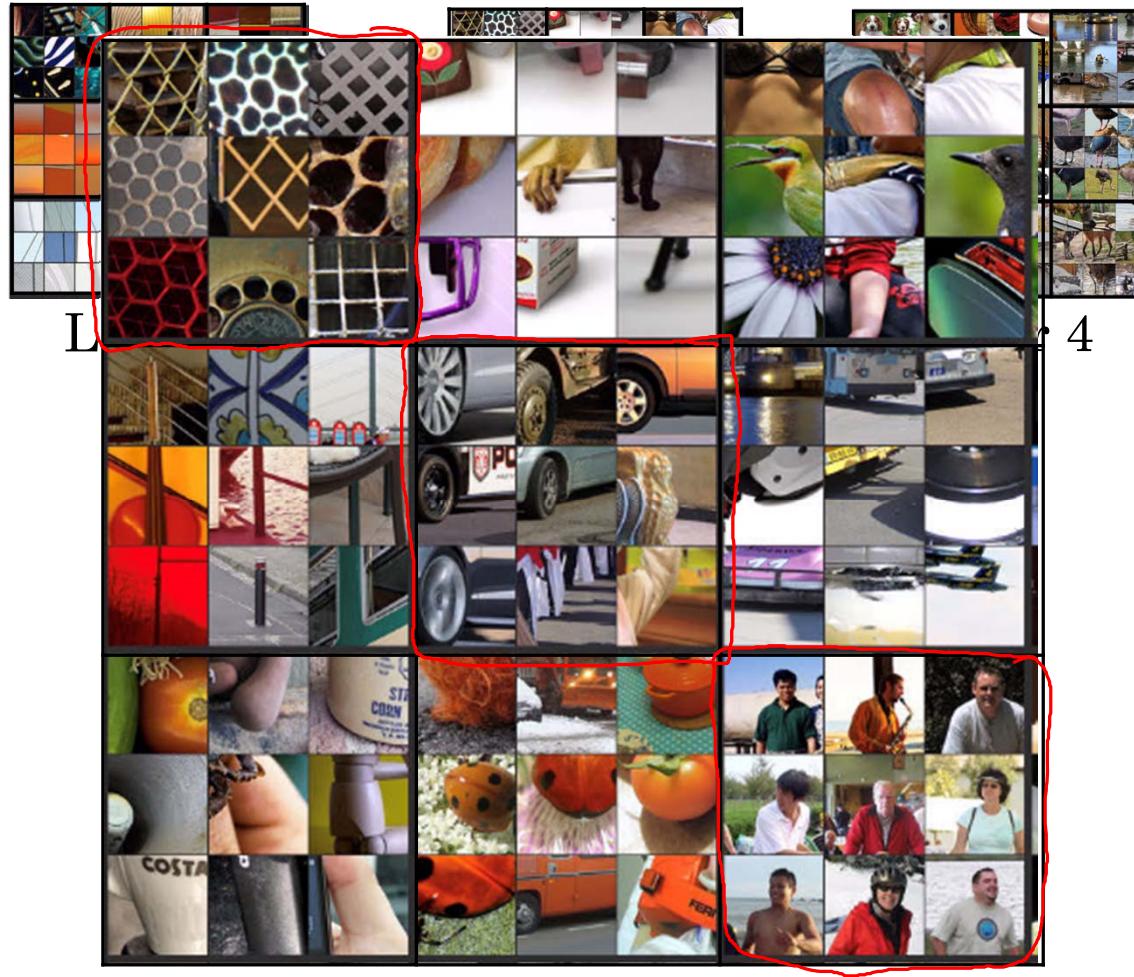


Andrew Ng

# Visualizing deep layers: Layer 3



Layer 1



Layer 5

Andrew Ng

# Visualizing deep layers: Layer 4

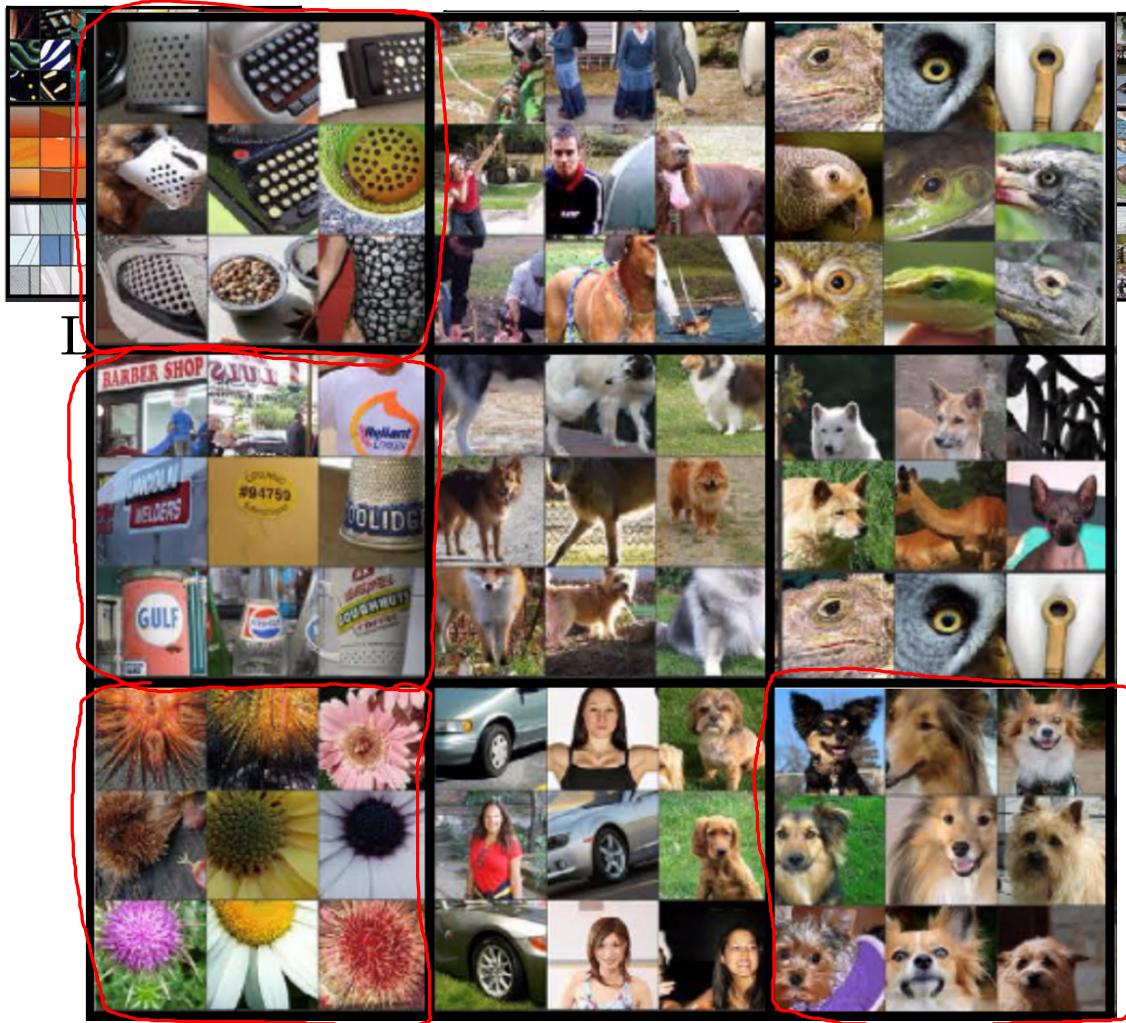


Andrew Ng

# Visualizing deep layers: Layer 5



Layer 1



Layer 5

Andrew Ng



deeplearning.ai

# Neural Style Transfer

---

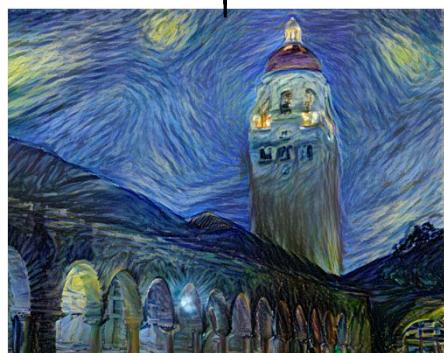
## Cost function

# Neural style transfer cost function



Content C

Style S



Generated image G

$$J(G) = \alpha J_{\text{Content}}(C, G)$$

$$+ \beta J_{\text{Style}}(S, G)$$

[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

$$G: \underbrace{100 \times 100}_{\text{RGB}} \times 3$$

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

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



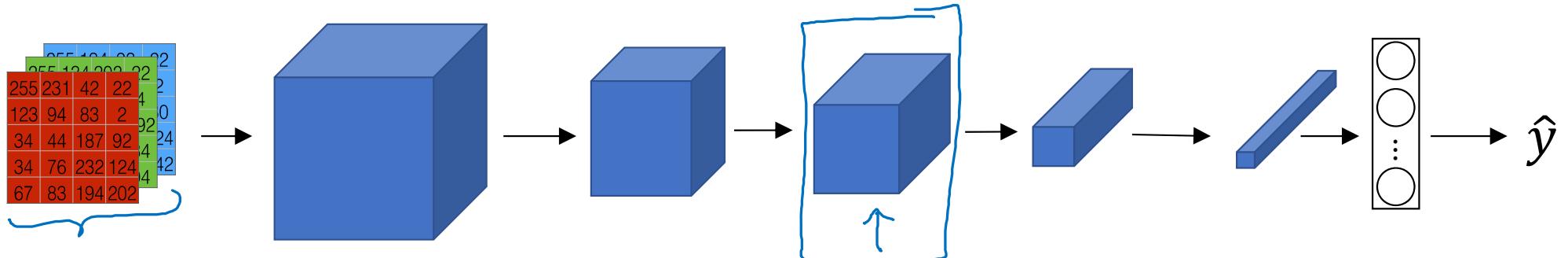
deeplearning.ai

# Neural Style Transfer

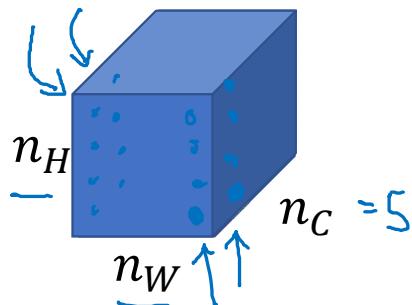
---

## Style cost function

# Meaning of the “style” of an image

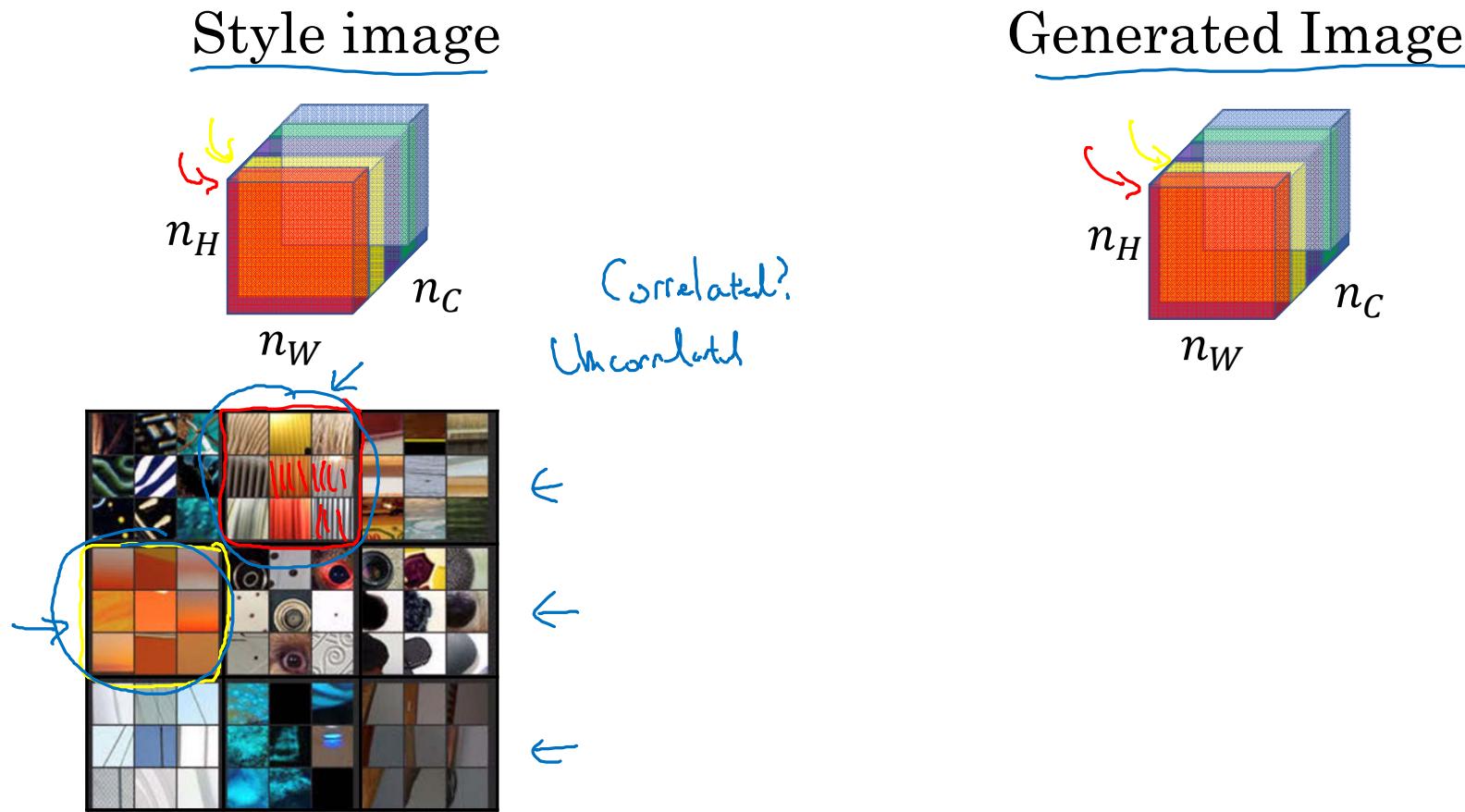


Say you are using layer  $l$ 's activation to measure “style.”  
Define style as correlation between activations across channels.



How correlated are the activations  
across different channels?

# Intuition about style of an 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  
↓

"Gram matrix"

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

$$\begin{aligned} J_{\text{style}}^{[l]}(S, G) &= \frac{1}{\binom{n_c}{2}} \| G^{[l](s)} - G^{[l](G)} \|_F^2 \\ &= \frac{1}{(2n_H^{[l]}n_W^{[l]}n_c^{[l]})^2} \sum_k \sum_{k'} (G_{kk'}^{[l](s)} - G_{kk'}^{[l](G)})^2 \end{aligned}$$

# Style cost function

$$\| G^{[l](s)} - G^{[l](G)} \|_F^2$$

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

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

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



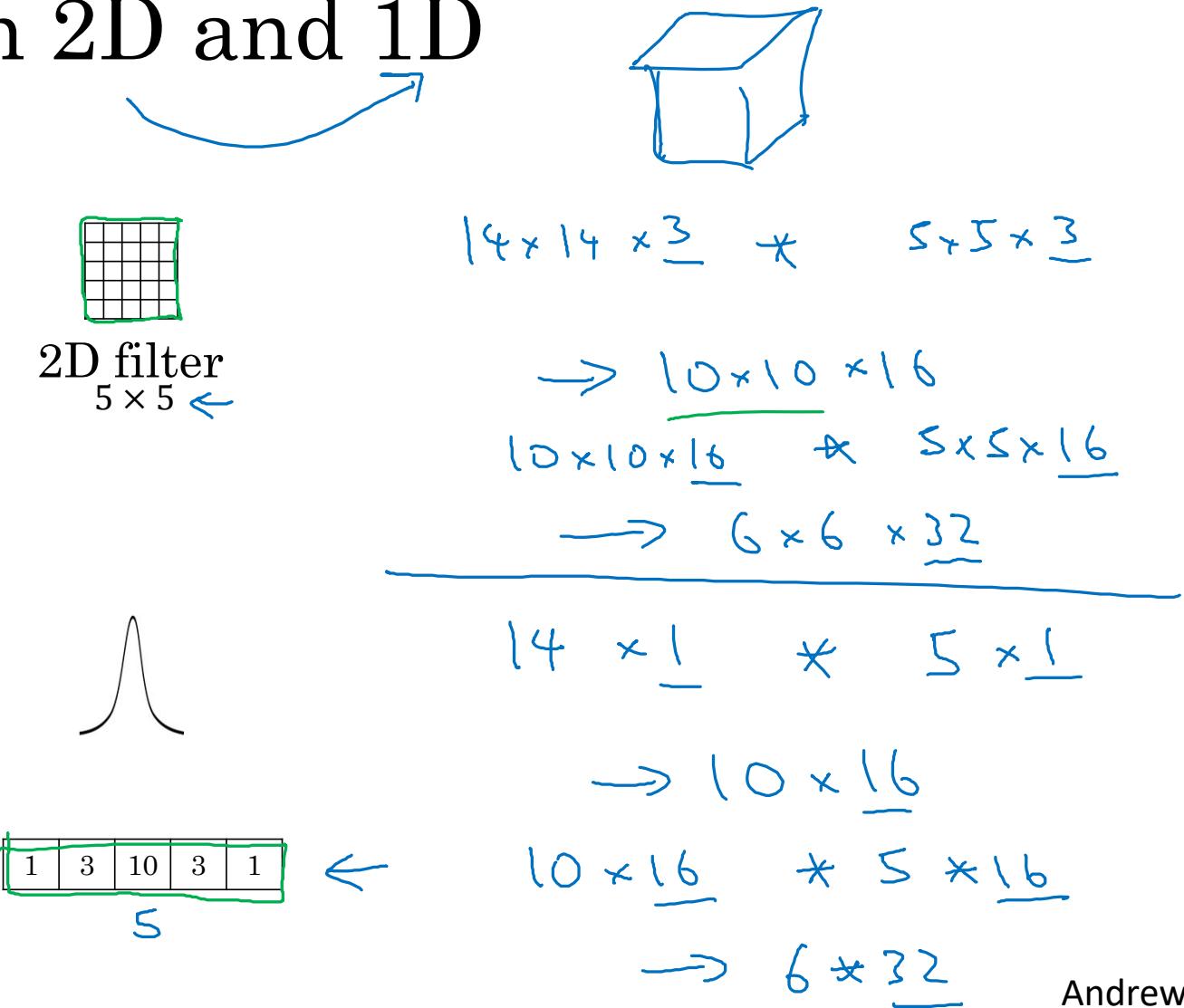
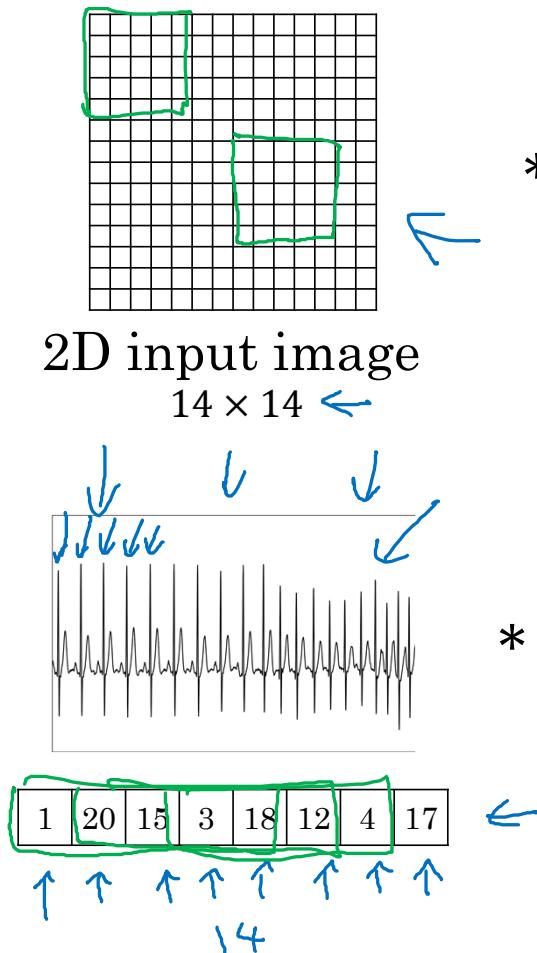
deeplearning.ai

# Convolutional Networks in 1D or 3D

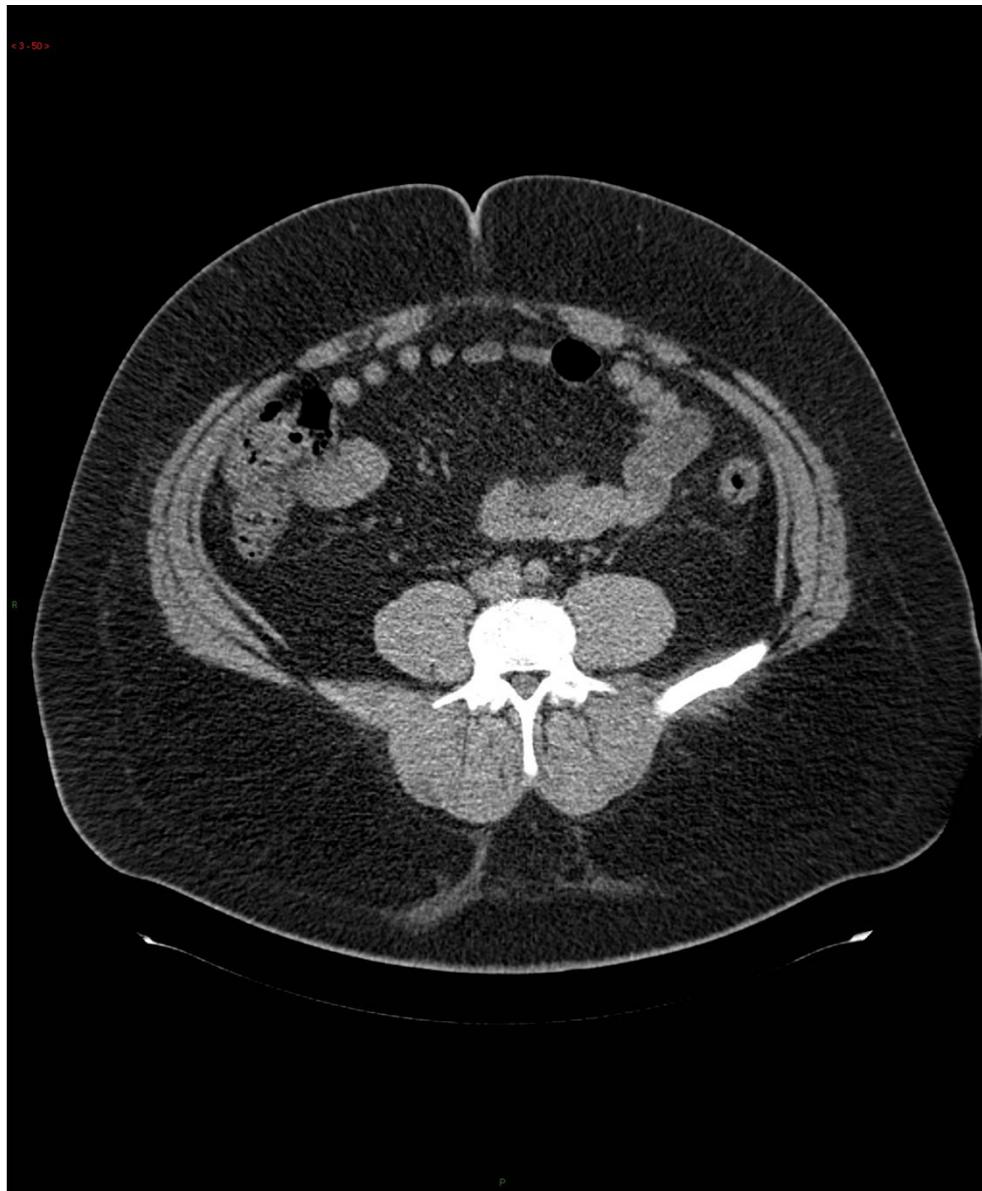
---

1D and 3D  
generalizations of  
models

# Convolutions in 2D and 1D



3D data



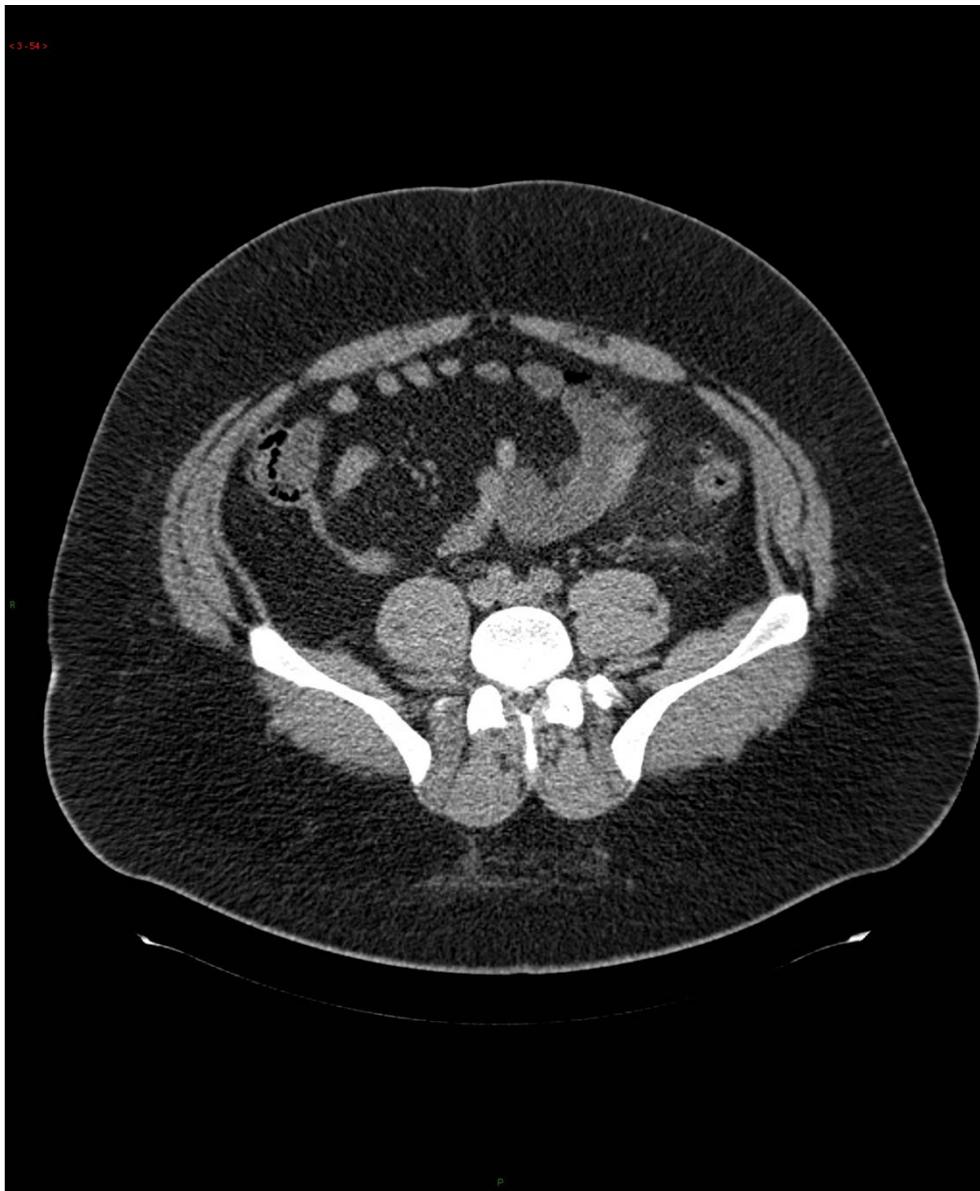
Andrew Ng

3D data



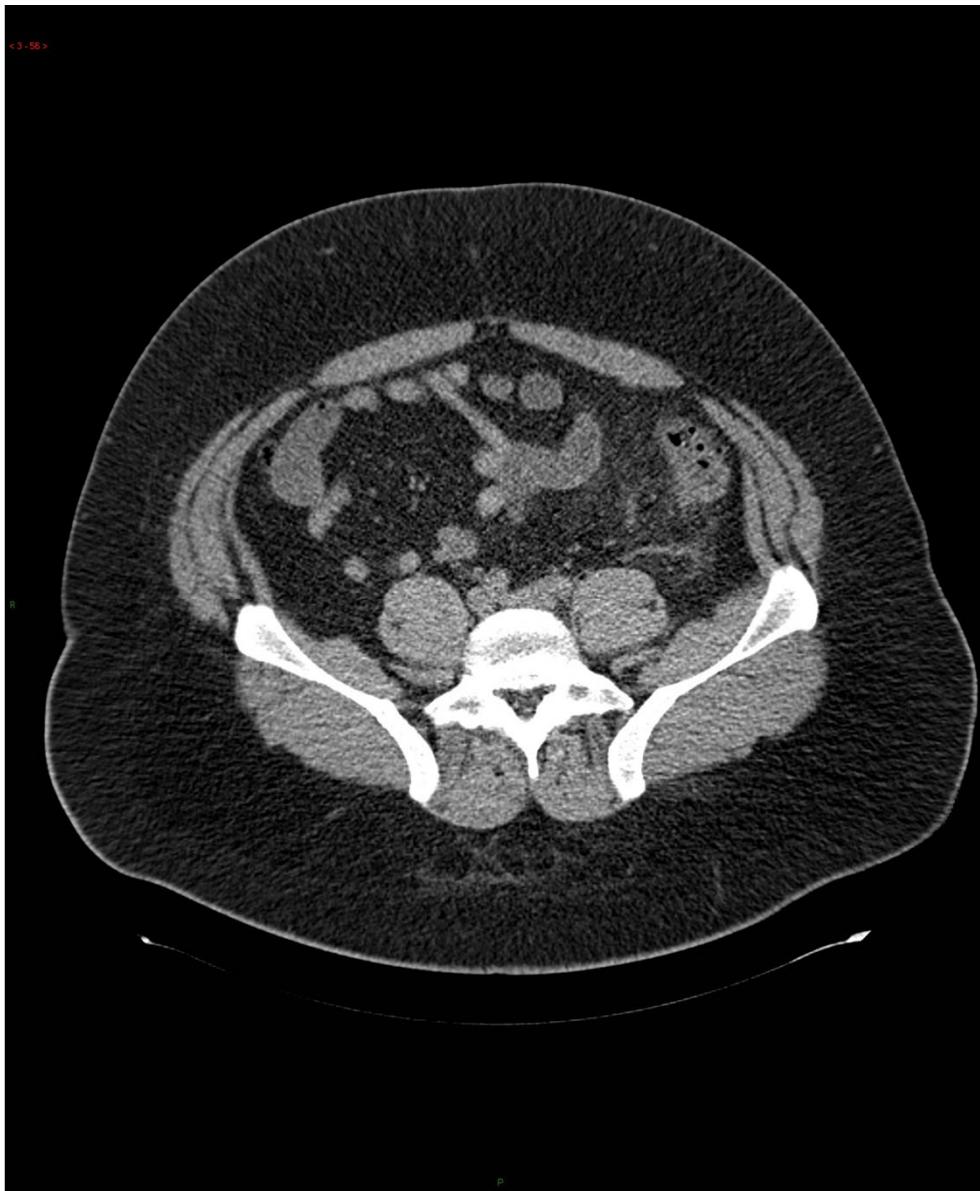
Andrew Ng

3D data



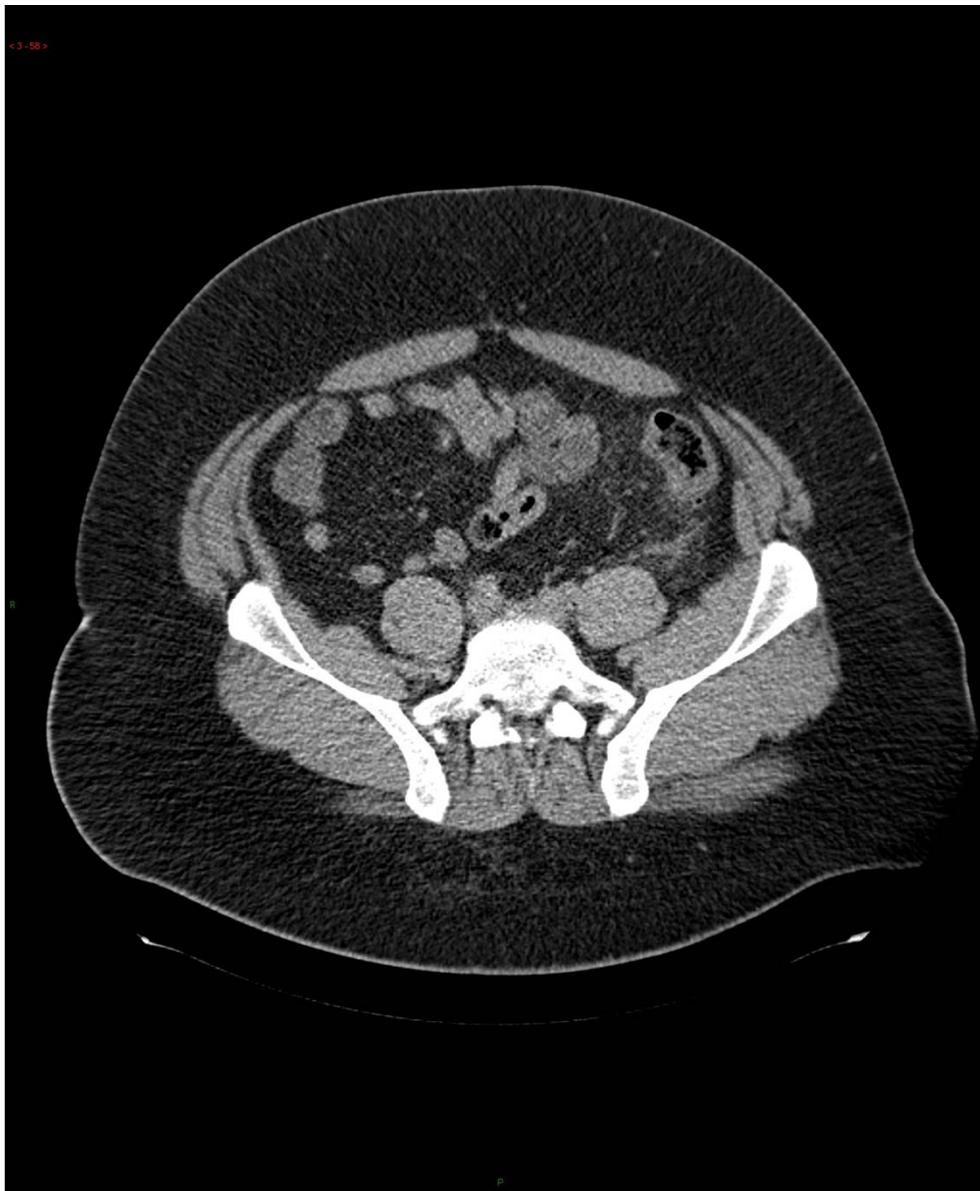
Andrew Ng

3D data



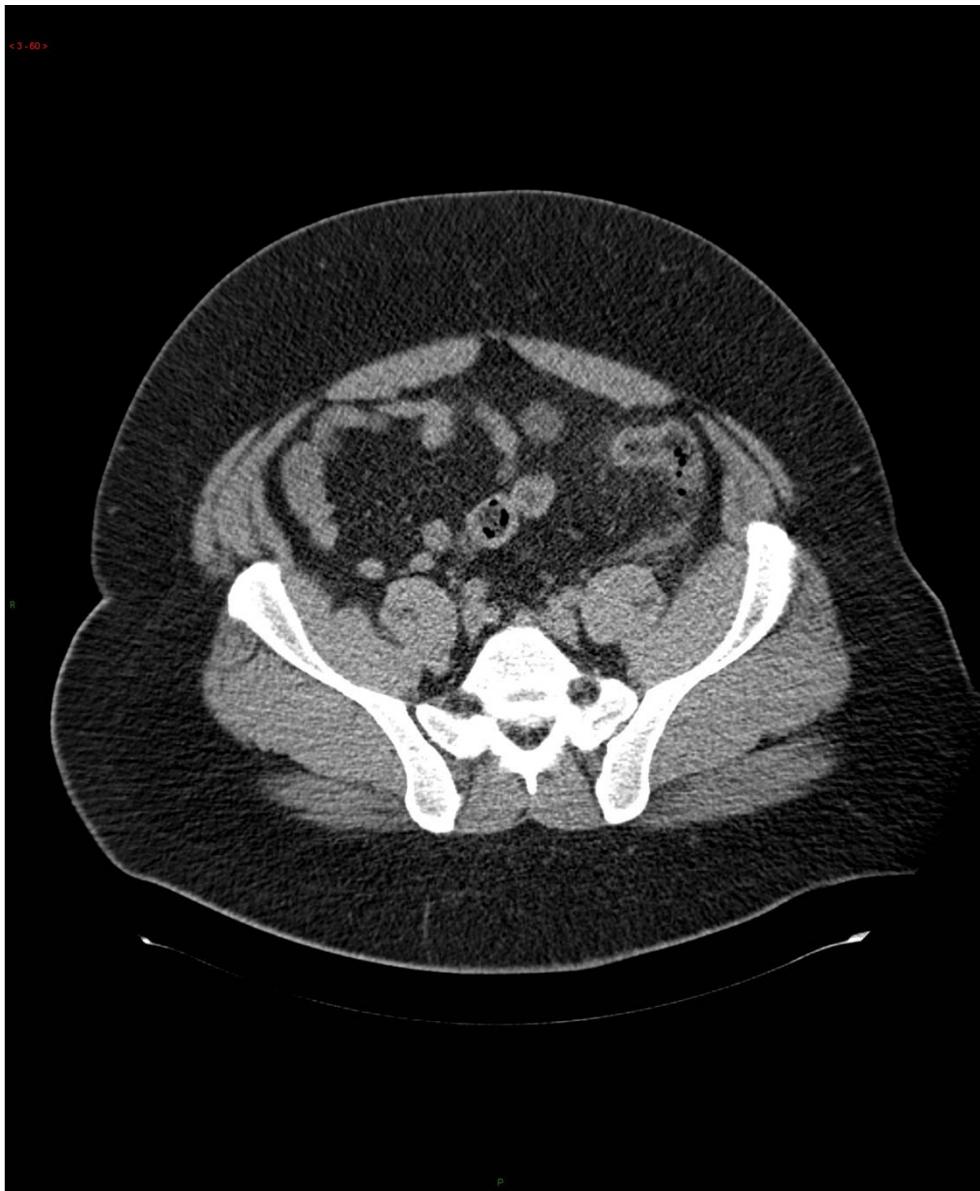
Andrew Ng

3D data



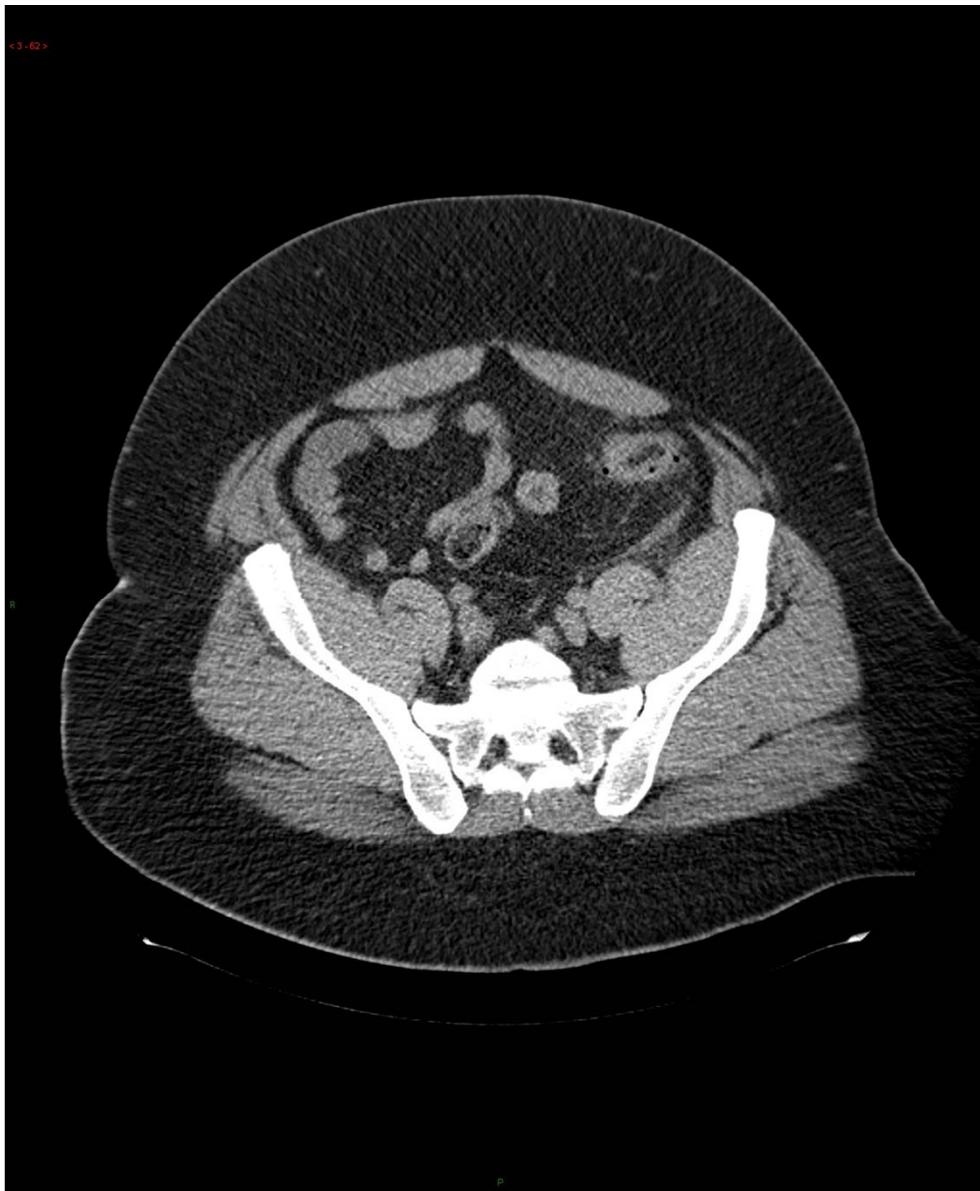
Andrew Ng

3D data



Andrew Ng

3D data



Andrew Ng

3D data



Andrew Ng

3D data



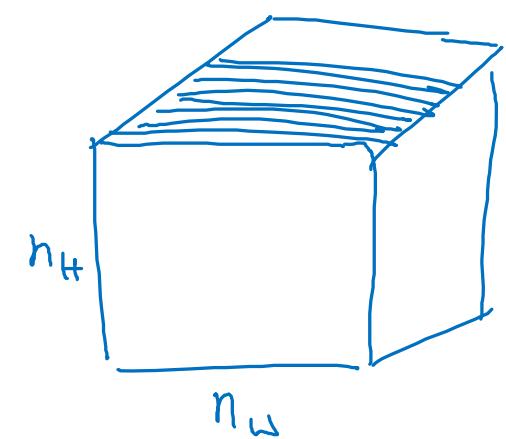
Andrew Ng

3D data



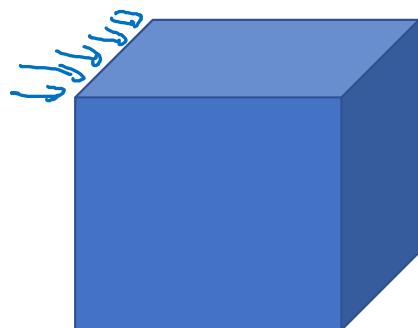
Andrew Ng

3D data

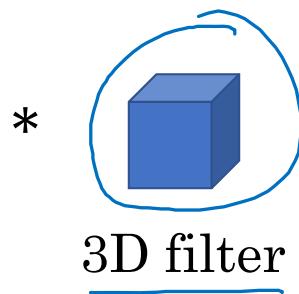


Andrew Ng

# 3D convolution



3D volume



$$\begin{array}{c} \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ | \quad 4 \times 14 \times 14 \quad \times 1 \\ \hline \quad \quad \quad \quad n_c \\ * \quad 5 \times 5 \times 5 \times 1 \\ \hline \rightarrow 10 \times 10 \times 10 \times 16 \\ * \quad 5 \times 5 \times 5 \times 16 \\ \hline \quad \quad \quad \quad 32 \text{ filters} \\ \rightarrow 6 \times 6 \times 6 \times 32 \end{array}$$

Andrew Ng