# Copyright Notice

deeplearning.ai

# Face recognition

## What is face recognition?

# Face recognition

[Courtesy of Baidu]

# Face verification vs. face recognition

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

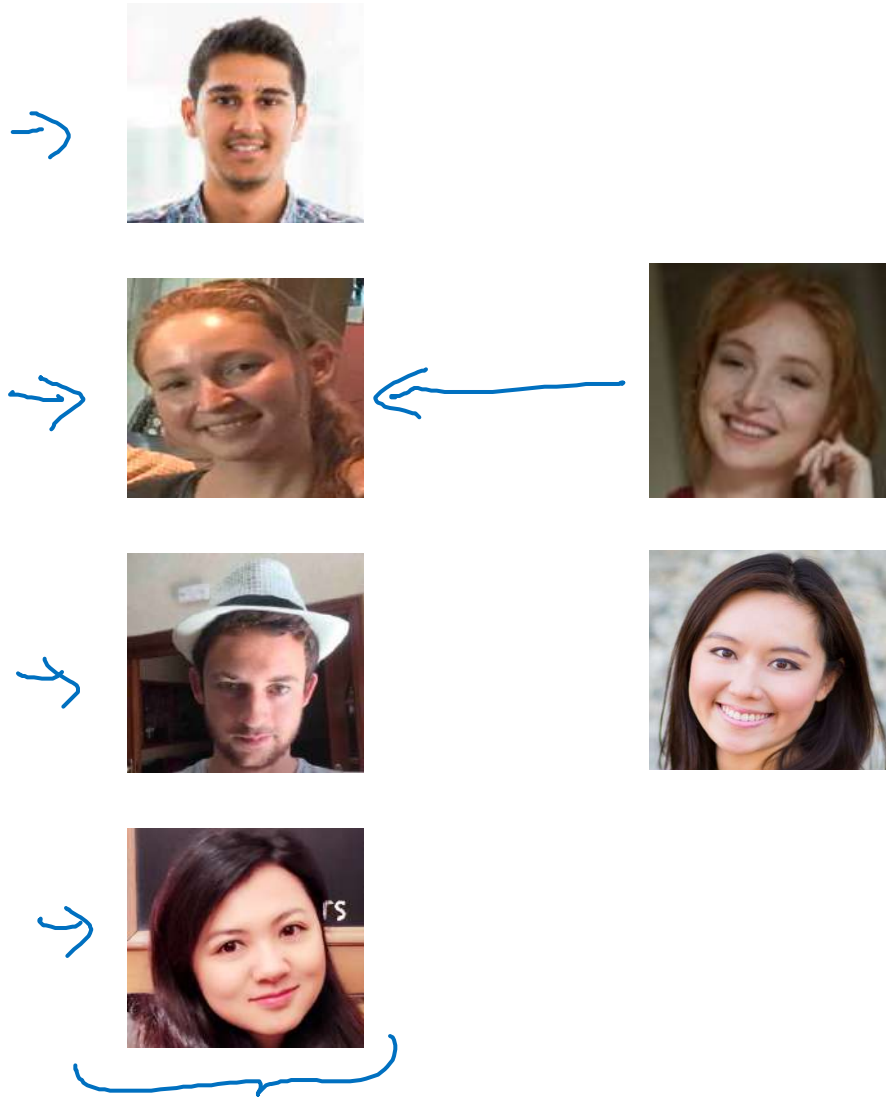
deeplearning.ai

# Face recognition

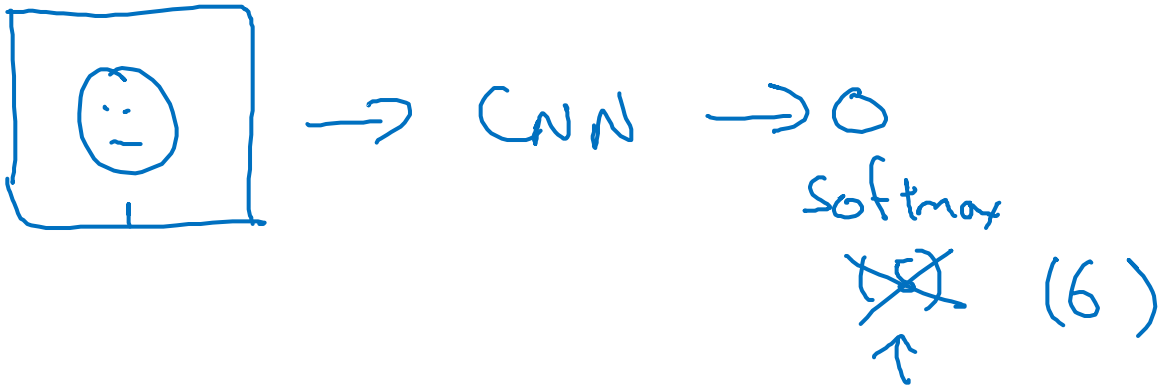
## One-shot learning

The difficult thing is that we will be having only 1 image of the person and DL technologies don't do w

# One-shot learning



Learning from one example to recognize the person again



# Learning a "similarity" function

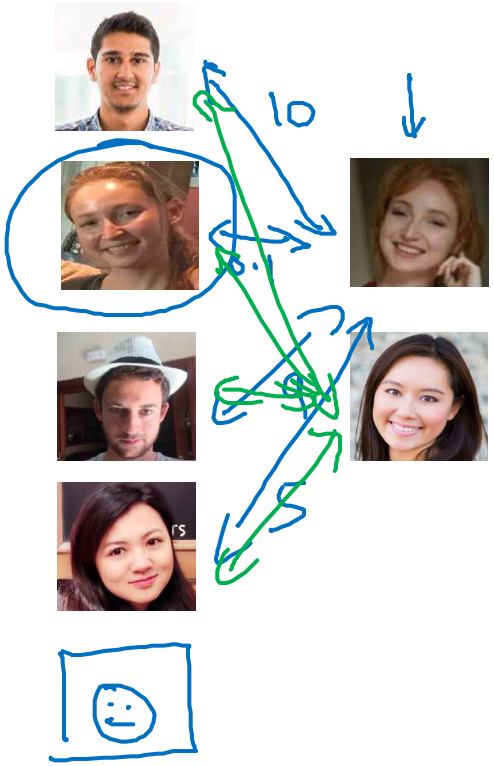
This function takes innput as two images and

→ d(img1,img2) = degree of difference between images

If d(img1,img2) $\leq \tau$ "same"

$> \tau$ "different"

} Verification.




d(img1, img2)

deeplearning.ai

# Face recognition

## Siamese network

# Siamese network

$x^{(1)}$

"encoding of $x^{(1)}$"

128

$f(x^{(1)})$

128

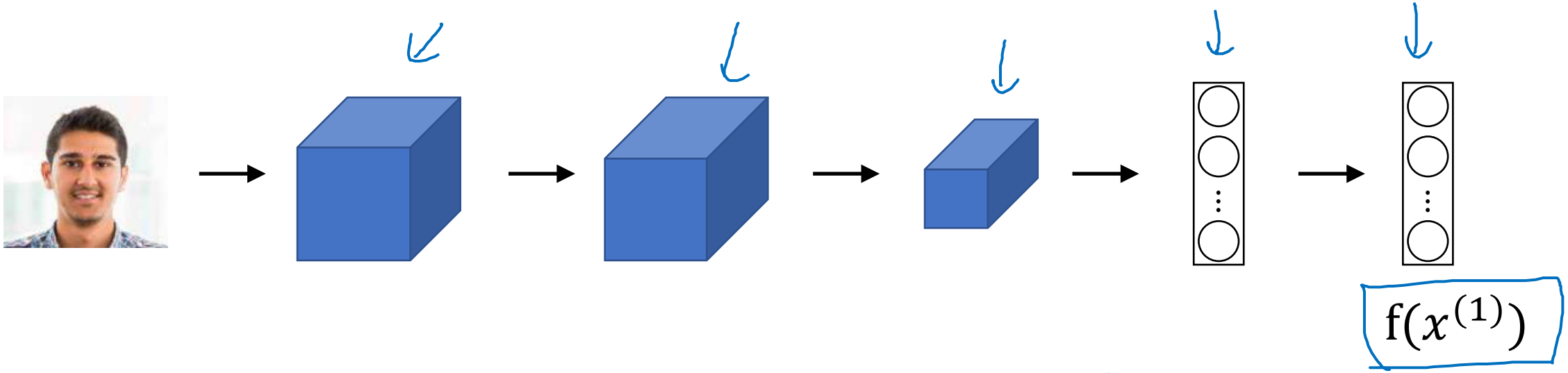
$x^{(2)}$

$f(x^{(2)})$

$$d(x^{(1)}, x^{(2)}) = \| f(x^{(1)}) - f(x^{(2)}) \|_2^2$$

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

# Goal of learning




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

128

Thsi is a fixed soem of the last

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.

deeplearning.ai

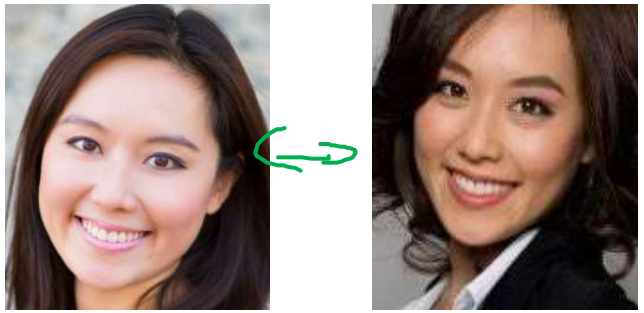
# Face recognition

## Triplet loss

Wo sab to theek hai lekin model ko kese train kare taaki bo last ke layers mein difference karne se si

# Learning Objective

Triplet nam aaya kyunki har ek time hum 3 images rakhenge. Ek anchor(o.e under fo





Anchor

Positive

Anchor

Negative

$d(A,P) \leq 0.5$

$d(A,N) = 0.7$

Want: $\|f(A) - f(P)\|^2 + \alpha \leq \|f(A) - f(N)\|^2$

$\alpha = 0.2$

$d(A,P)$ — $d(A,N)$ —

$$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha \leq 0$$

$\alpha$ = margin

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

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

# Loss function

Given 3 images A, P, N:

Triplet loss function.

$$\mathcal{L}(A, P, N) = \max\left(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0\right)$$

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

A, P

Training set: 10k pictures of 1k persons

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

# 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.

$\|f(A) - f(P)\|^2 + \alpha \leq \|f(A) - f(N)\|^2$

Choose triplets that're "hard" to train on.

$d(A,P) + \alpha \leq d(A,N)$

$d(A,P) \approx d(A,N)$

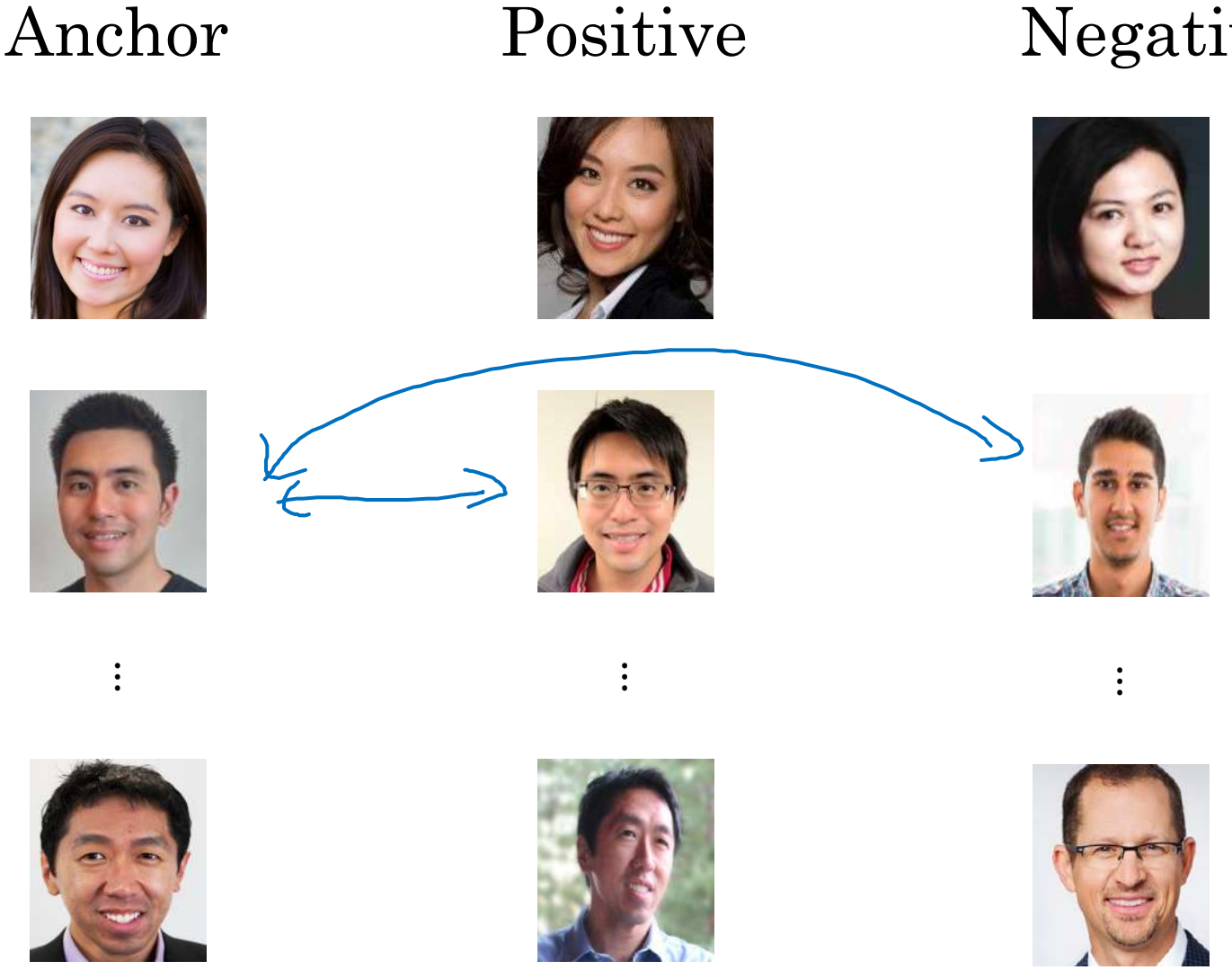
Face Net

Deep Face

Hard.

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

# Training set using triplet loss




$J$

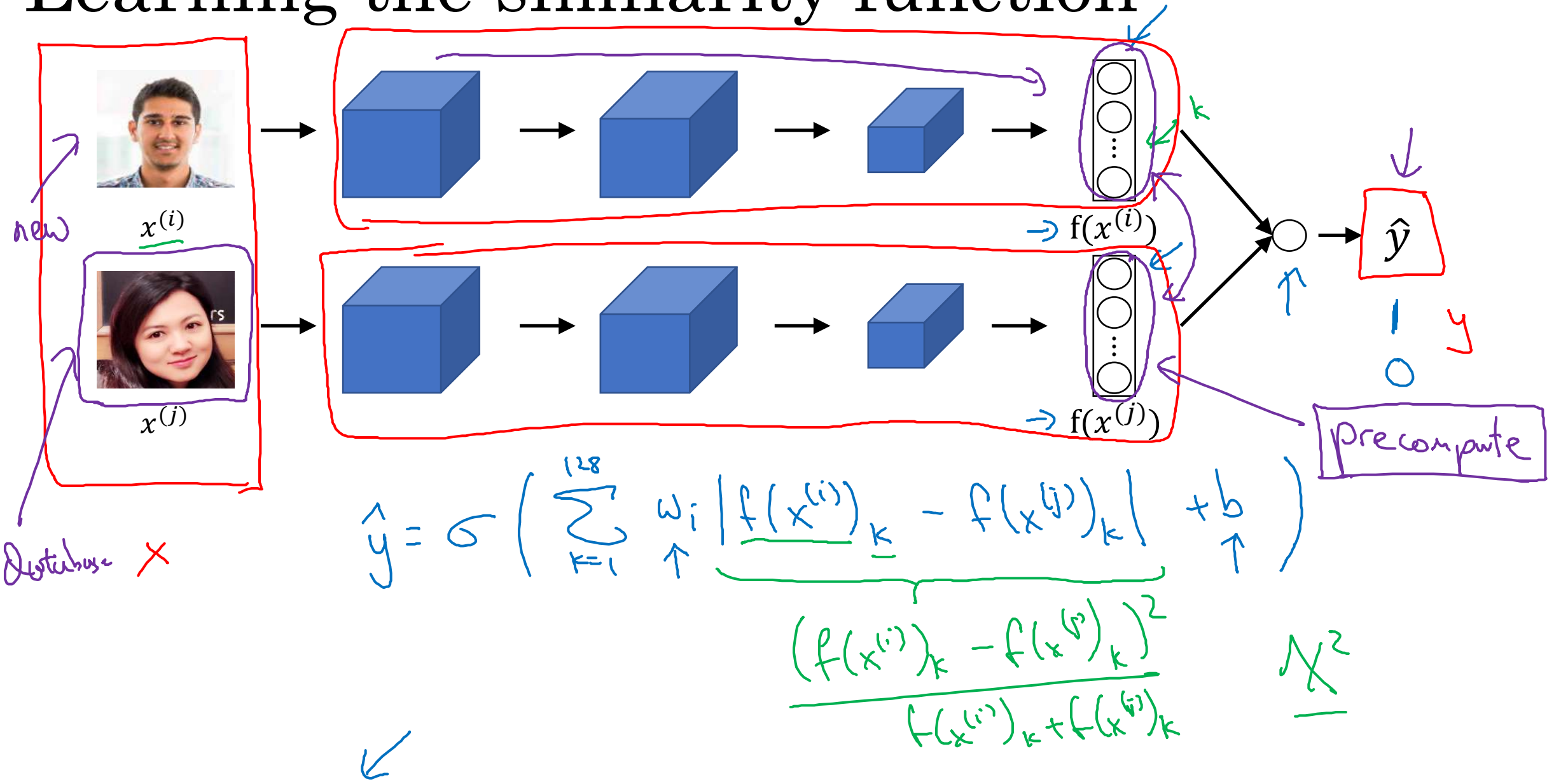
$d(x^{(i)}, x^{(j)})$

deeplearning.ai

# Face recognition

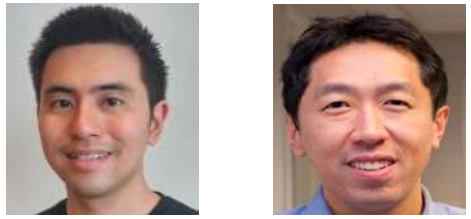
# Face verification and binary classification

# Learning the similarity function




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

# Face verification supervised learning

| $x$ | $y$ | |
|---|---|---|
|  | 1 | "Same" |
|  | 0 | "Albert" |
|  | 0 | |
|  | 1 | |

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

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)

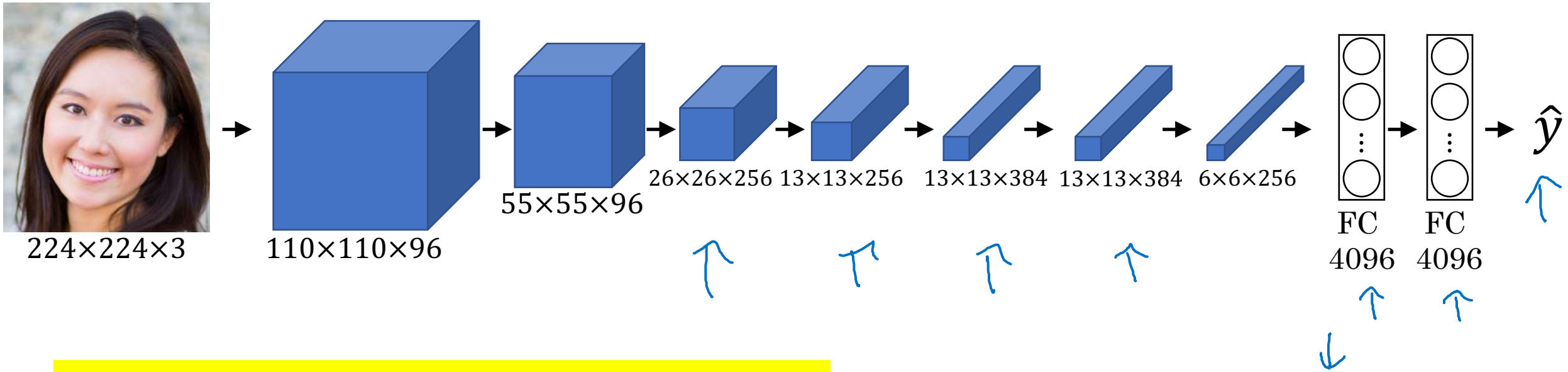
[Images generated by Justin Johnson]

deeplearning.ai

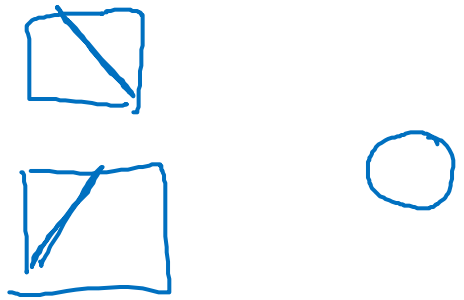
# Neural Style Transfer

## What are deep ConvNets learning?

# Visualizing what a deep network is learning



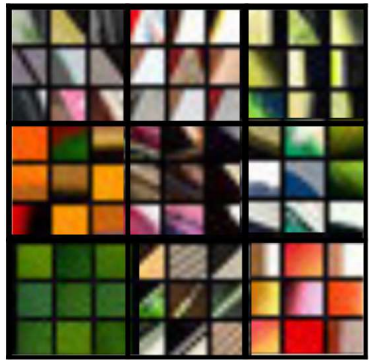

Pick a unit in layer 1. Find the nine image patches that maximize the unit's activation.



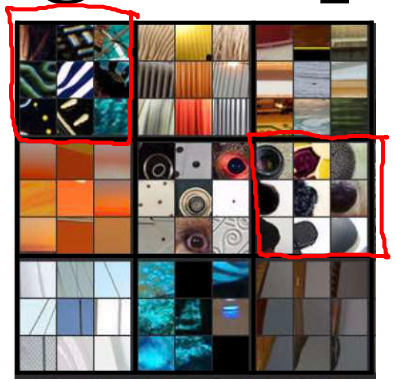
Repeat for other units.

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

# Visualizing deep layers

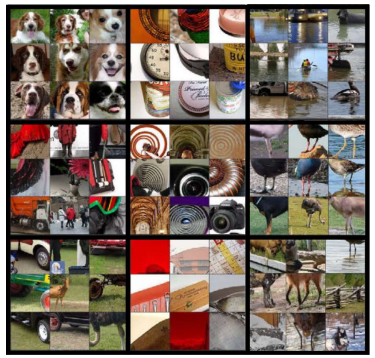


Layer 1

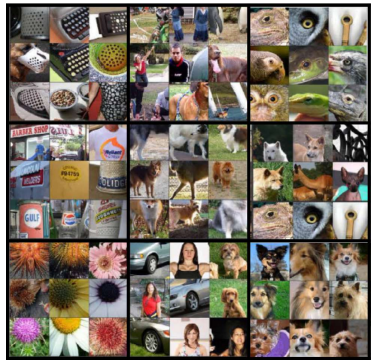


Layer 2

Layer 3

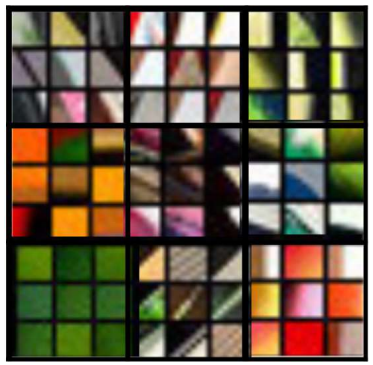


Layer 4



Layer 5

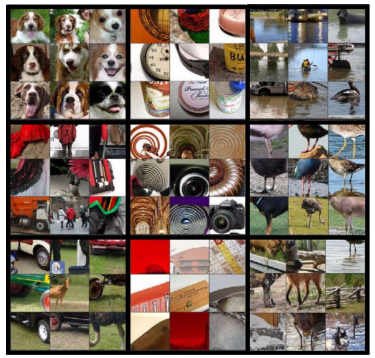
# Visualizing deep layers: Layer 1



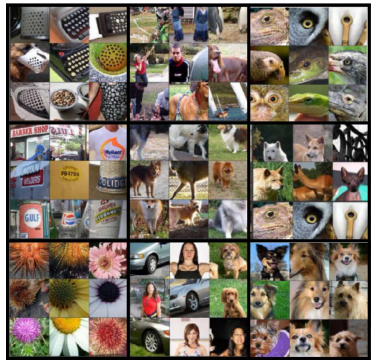
Layer 1

Layer 2

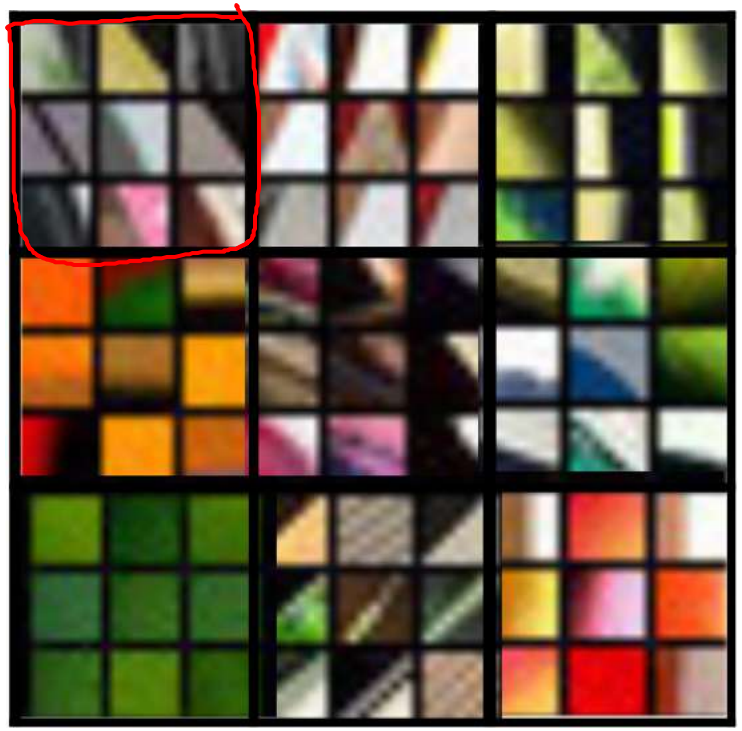
Layer 3



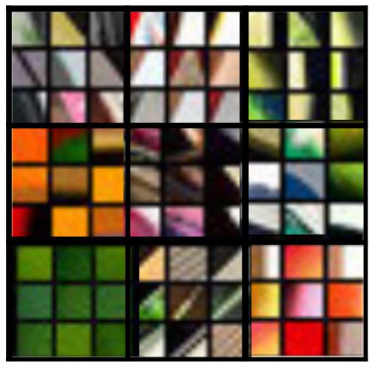
Layer 4



Layer 5



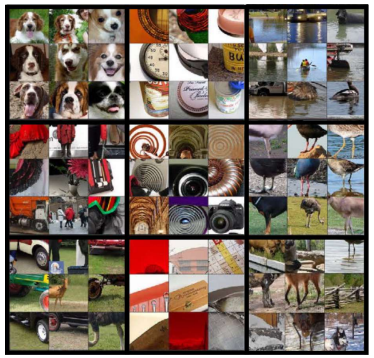
# Visualizing deep layers: Layer 2



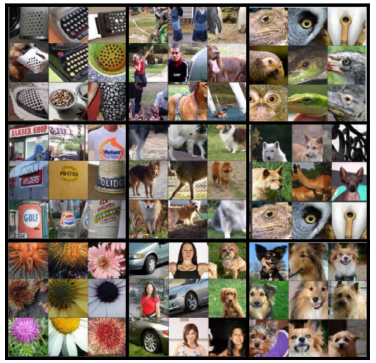
Layer 1

Layer 2

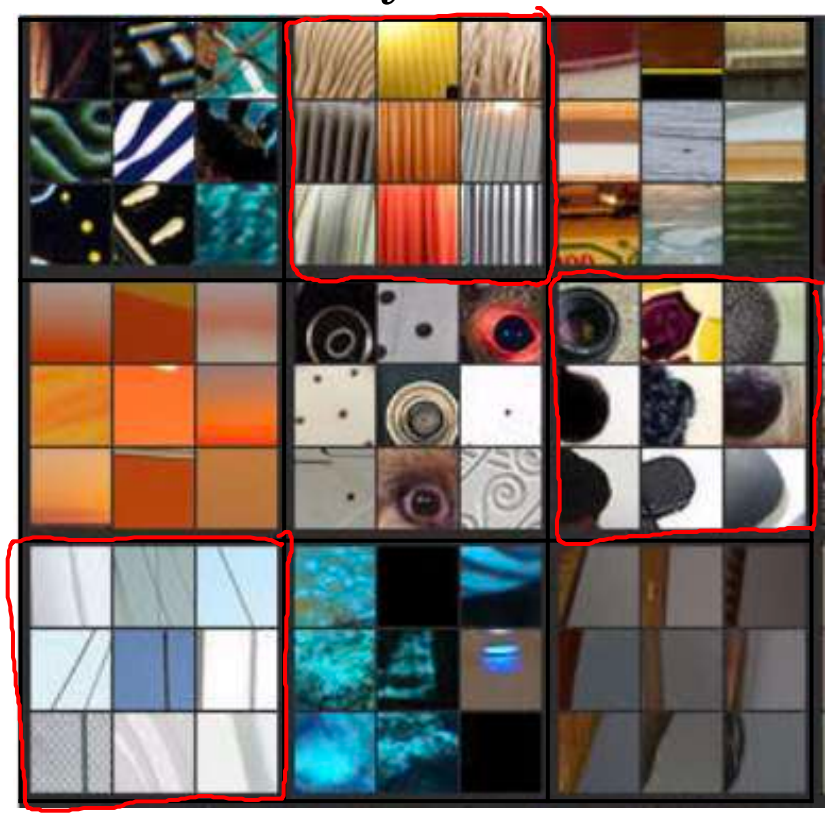
Layer 3



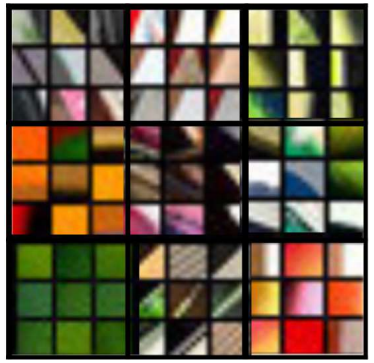
Layer 4



Layer 5



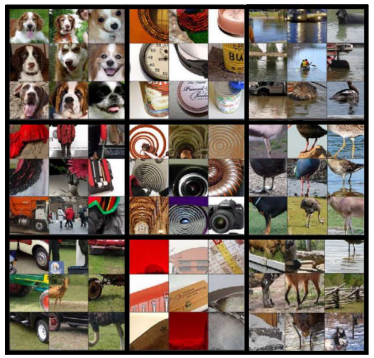
# Visualizing deep layers: Layer 3



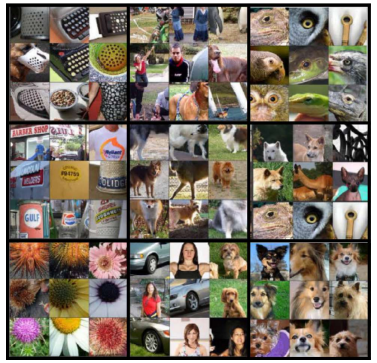
Layer 1

Layer 2

Layer 3

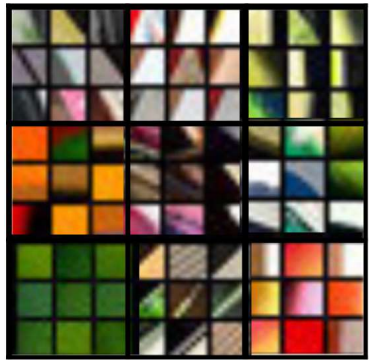


Layer 4

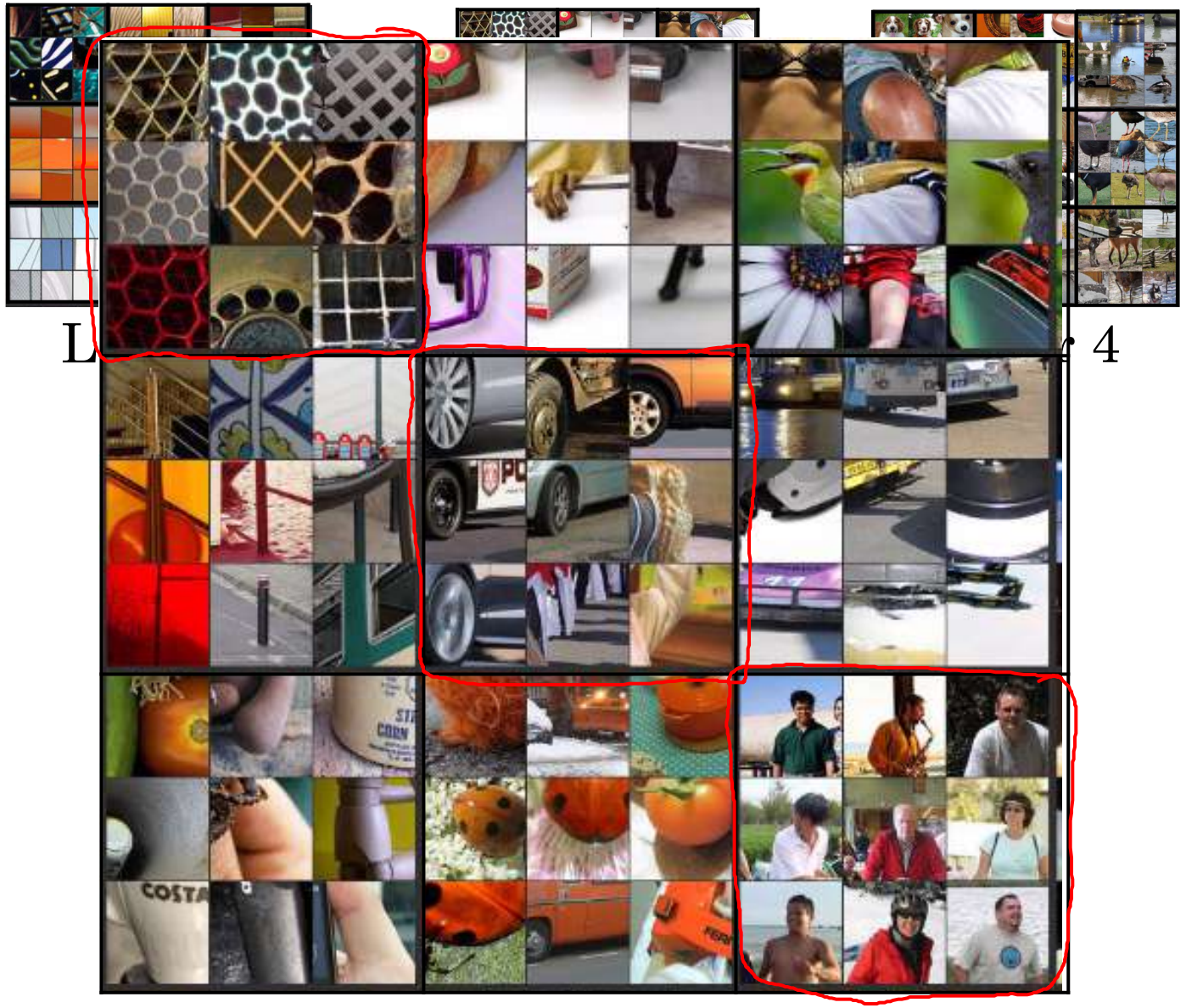


Layer 5

# Visualizing deep layers: Layer 3

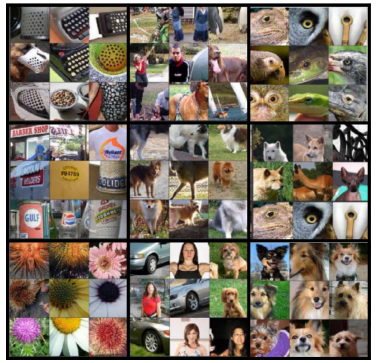


Layer 1



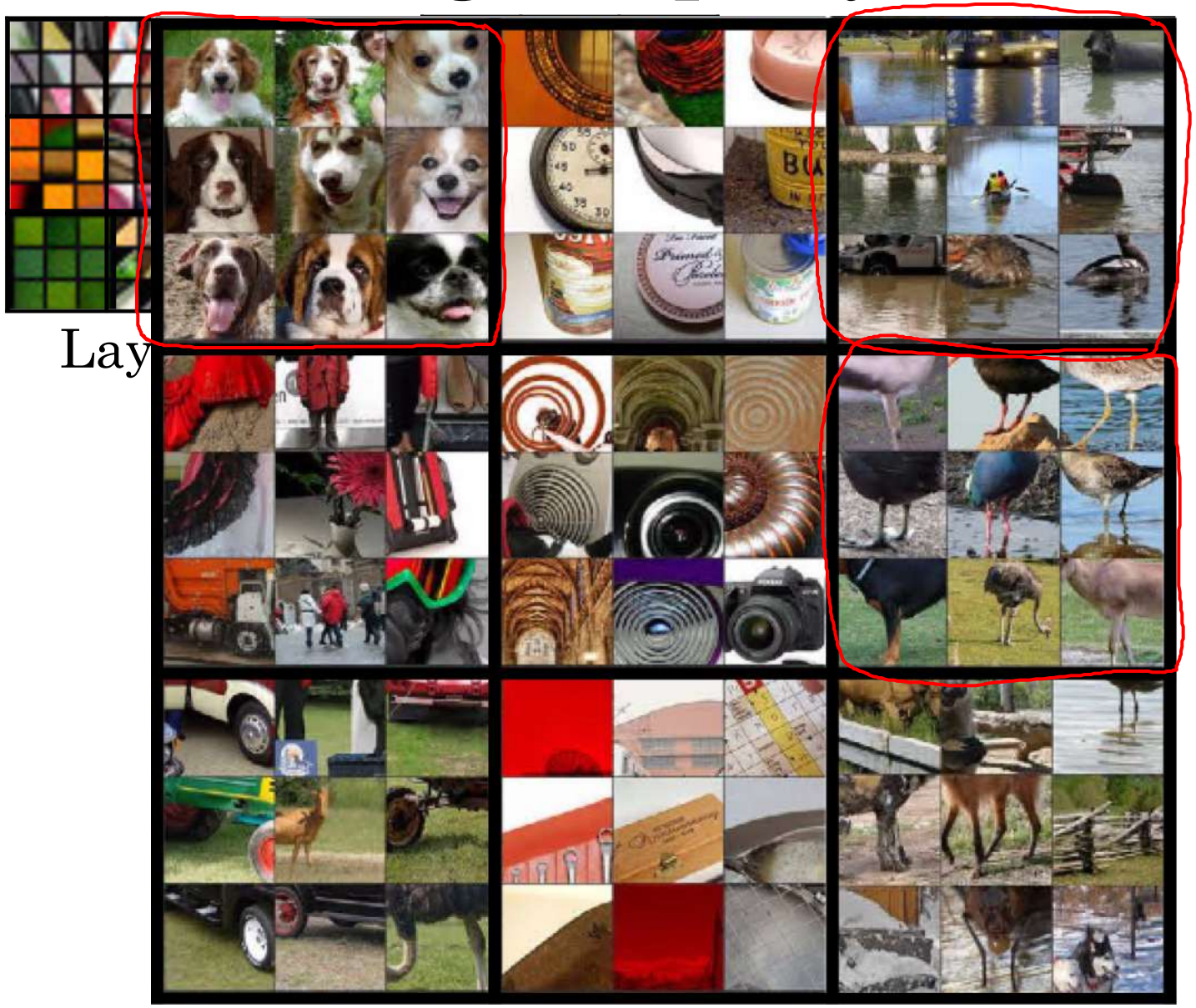
L

4

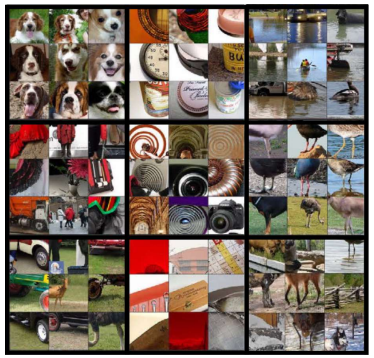


Layer 5

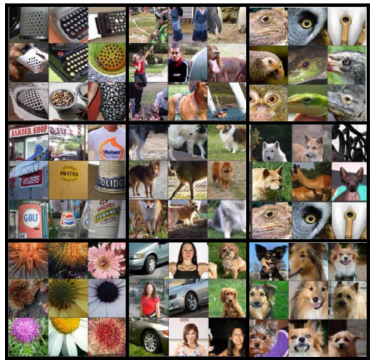
# Visualizing deep layers: Layer 4



Lay

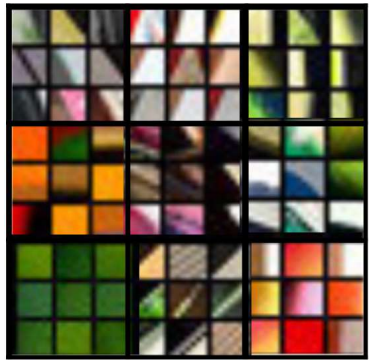


Layer 4

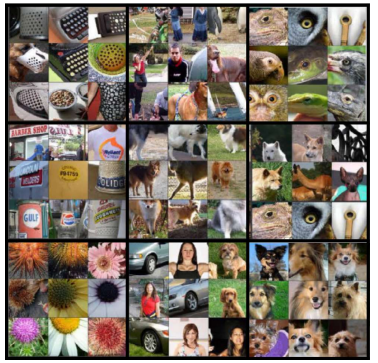


Layer 5

# Visualizing deep layers: Layer 5



Layer 1

Layer 5

Andrew Ng

deeplearning.ai

# Neural Style Transfer

## Cost function

# Neural style transfer cost function

J(C,G) defines how similar is conter

Content C

Style S

Generated image G

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

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

# Find the generated image G

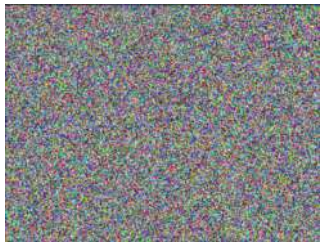
1. Initiate G randomly

   G: 100×100×3 (RGB)

   or whatever dimension we want.

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]

deeplearning.ai

# Neural Style Transfer

## Content cost function

# Content cost function

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

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

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

Somewhare in betwe

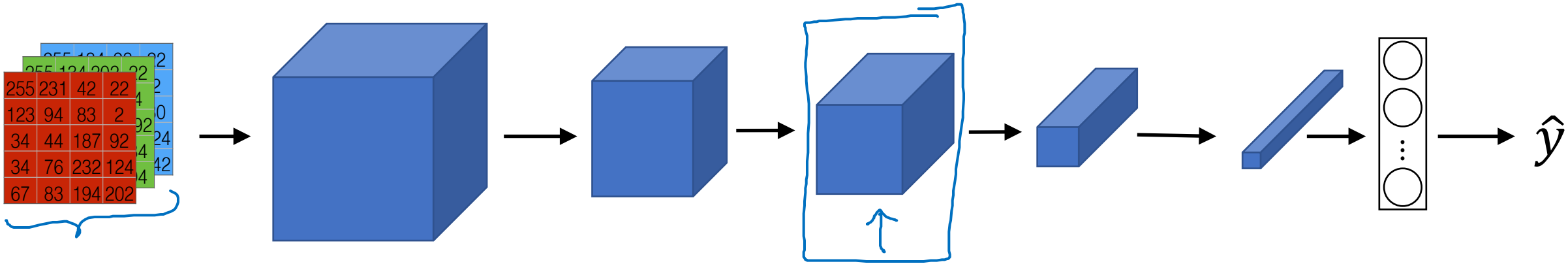
[Gatys et al., 2015. A neural algorithm of artistic style]

deeplearning.ai

# Neural Style Transfer

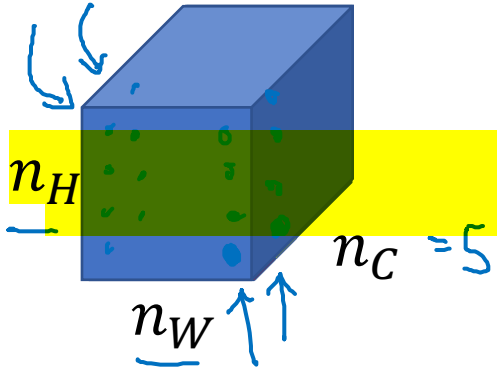
## Style cost function

# 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.

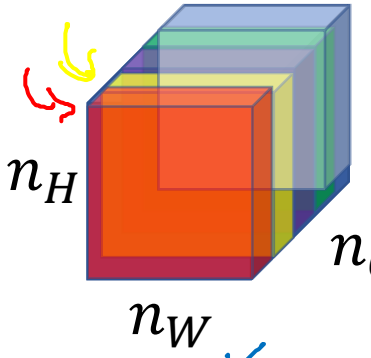



How correlated are the activations across different channels?

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

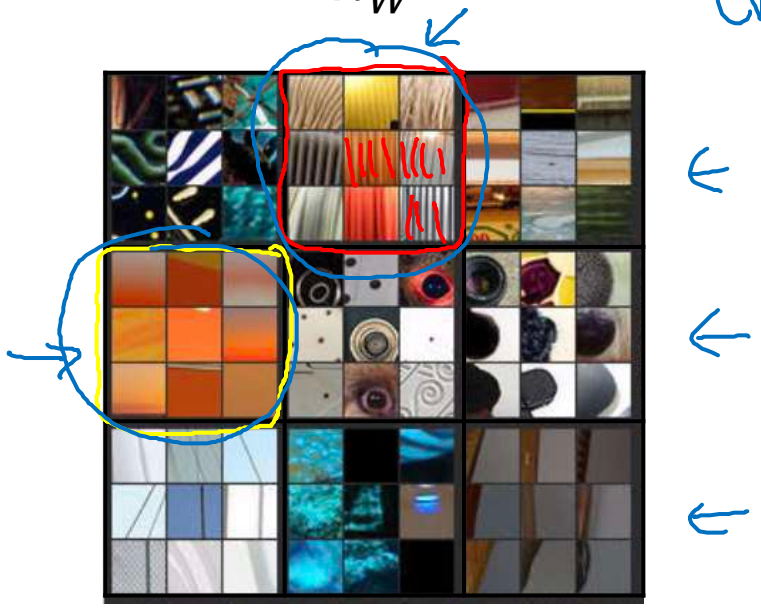
# Intuition about style of an image

## Style image

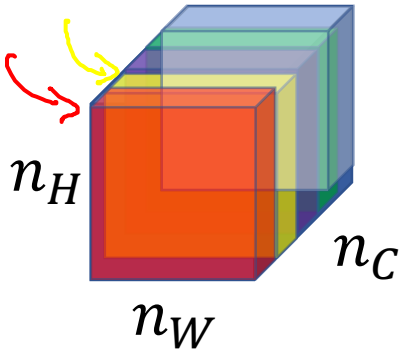



Correlated?

Uncorrelated



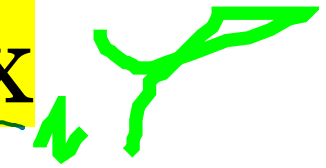
## Generated Image




More correlated means they are both closely related.

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

# Style matrix



H W C

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

$n_c$

$G^{[l]}_{kk'}$

$k = 1, \ldots, n_c^{[l]}$

$$G^{[l](S)}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a^{[l](S)}_{ijk} \, a^{[l](S)}_{ijk'}$$

$$G^{[l](G)}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a^{[l](G)}_{ijk} \, a^{[l](G)}_{ijk}$$

"Gram matrix"

$$J^{[l]}_{style}(S,G) = \frac{1}{(\ldots)} \left\| G^{[l](S)} - G^{[l](G)} \right\|_F^2$$

$$= \frac{1}{(2 n_H^{[l]} n_W^{[l]} n_c^{[l]})^2} \sum_k \sum_{k'} \left( G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'} \right)^2$$

$\beta$

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

# Style cost function

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

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

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

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

$G$

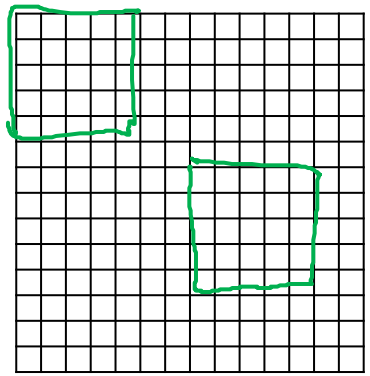
[Gatys et al., 2015. A neural algorithm of artistic style]

deeplearning.ai

# Convolutional Networks in 1D or 3D

## 1D and 3D generalizations of models

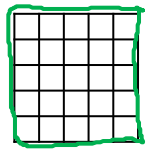
# Convolutions in 2D and 1D



2D input image

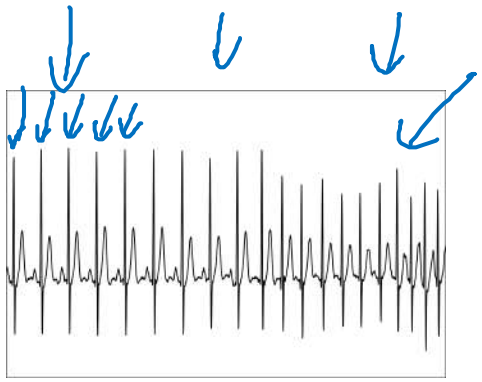
14×14

*

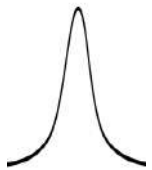


2D filter

5×5



*

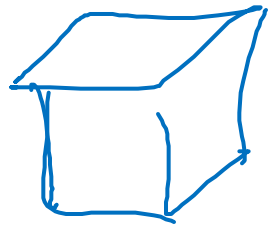


| 1 | 20 | 15 | 3 | 18 | 12 | 4 | 17 |
|---|---|---|---|---|---|---|---|

14

| 1 | 3 | 10 | 3 | 1 |
|---|---|---|---|---|

5



$14 \times 14 \times 3 \quad * \quad 5 \times 5 \times 3$

$\rightarrow 10 \times 10 \times 16$

$10 \times 10 \times 16 \quad * \quad 5 \times 5 \times 16$

$\rightarrow 6 \times 6 \times 32$

---

$14 \times 1 \quad * \quad 5 \times 1$

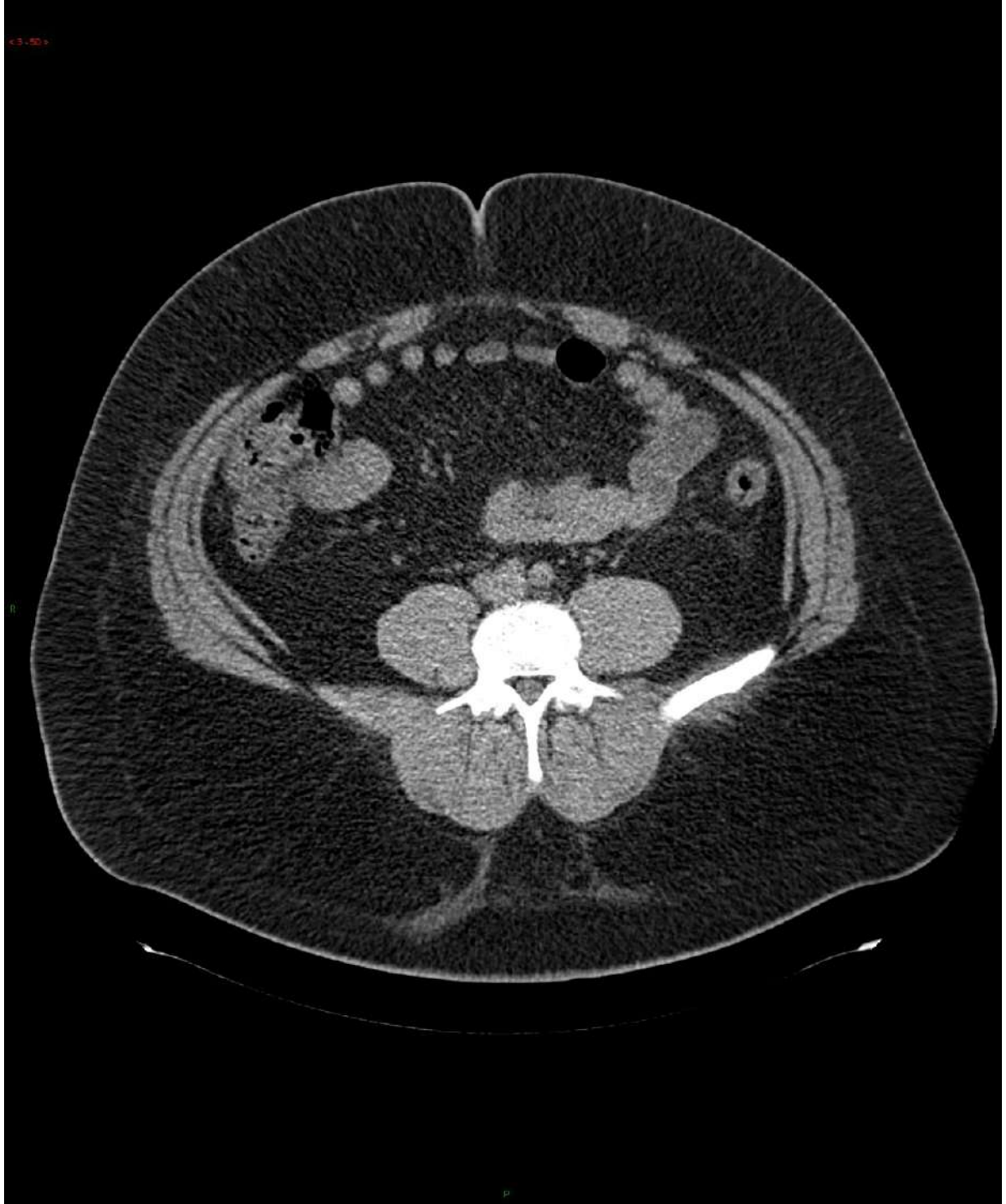
$\rightarrow 10 \times 16$

$10 \times 16 \quad * \quad 5 \times 16$

$\rightarrow 6 \times 32$

Andrew Ng

# 3D data

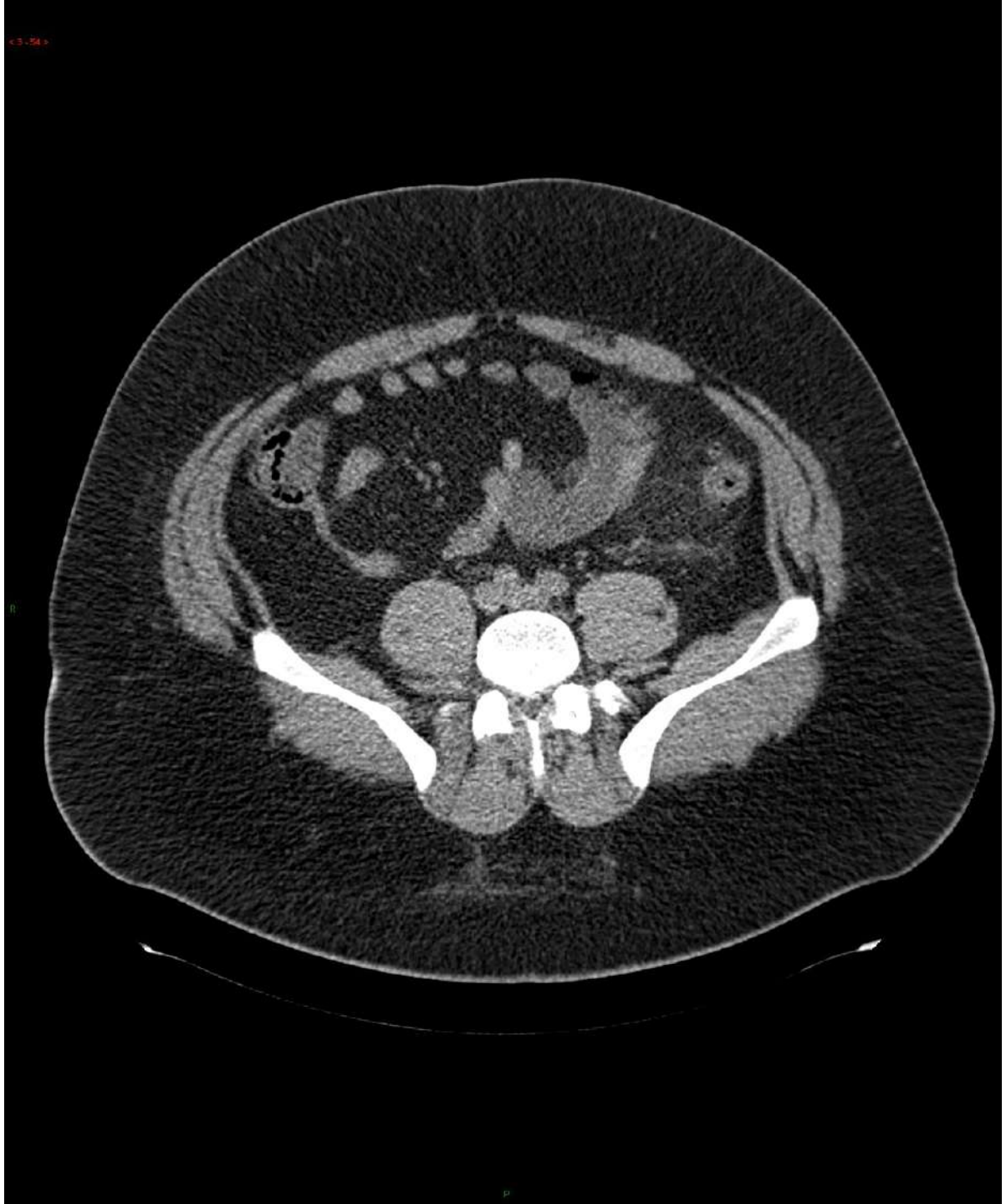


Andrew Ng

# 3D data

Andrew Ng

# 3D data

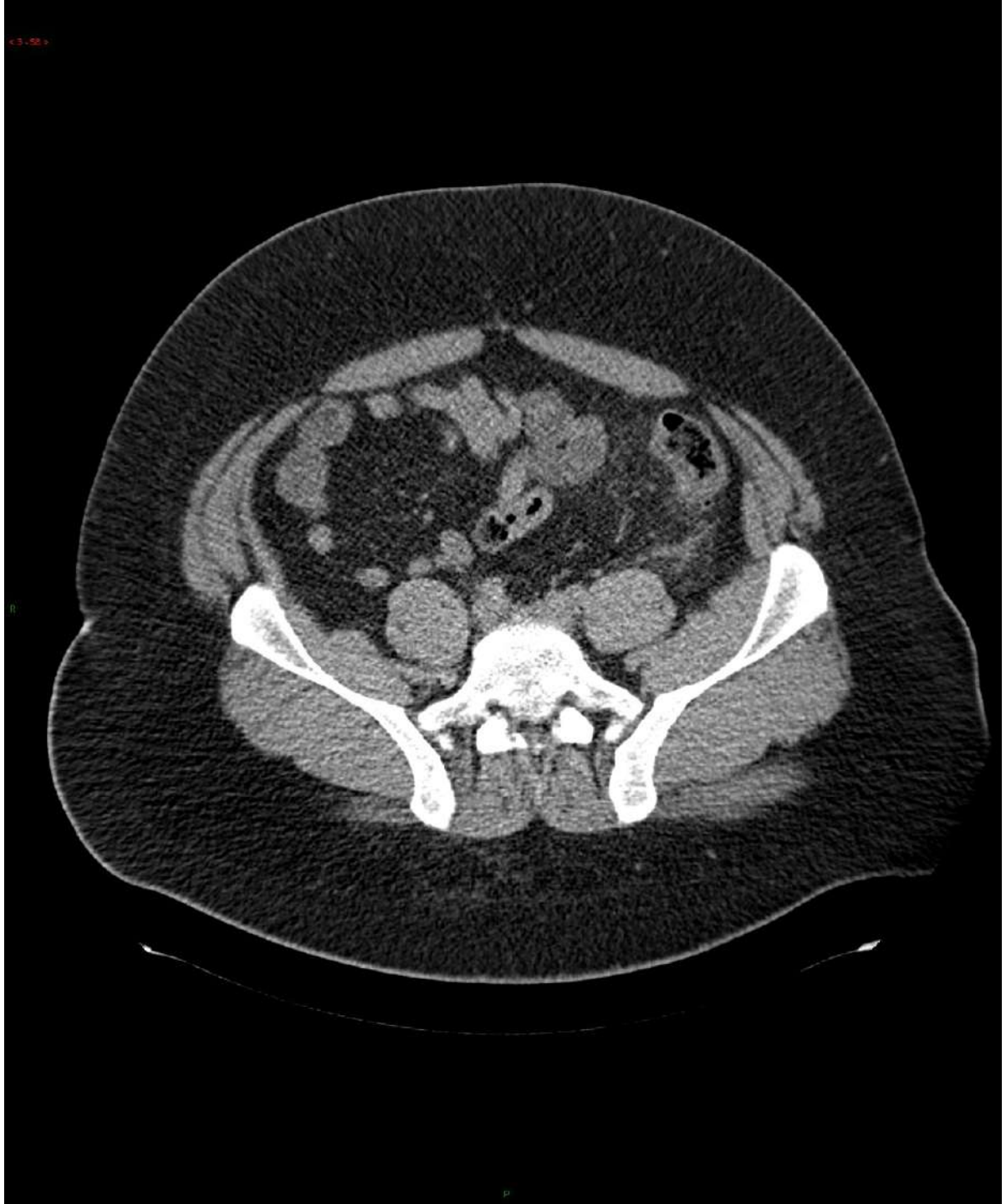


Andrew Ng

# 3D data

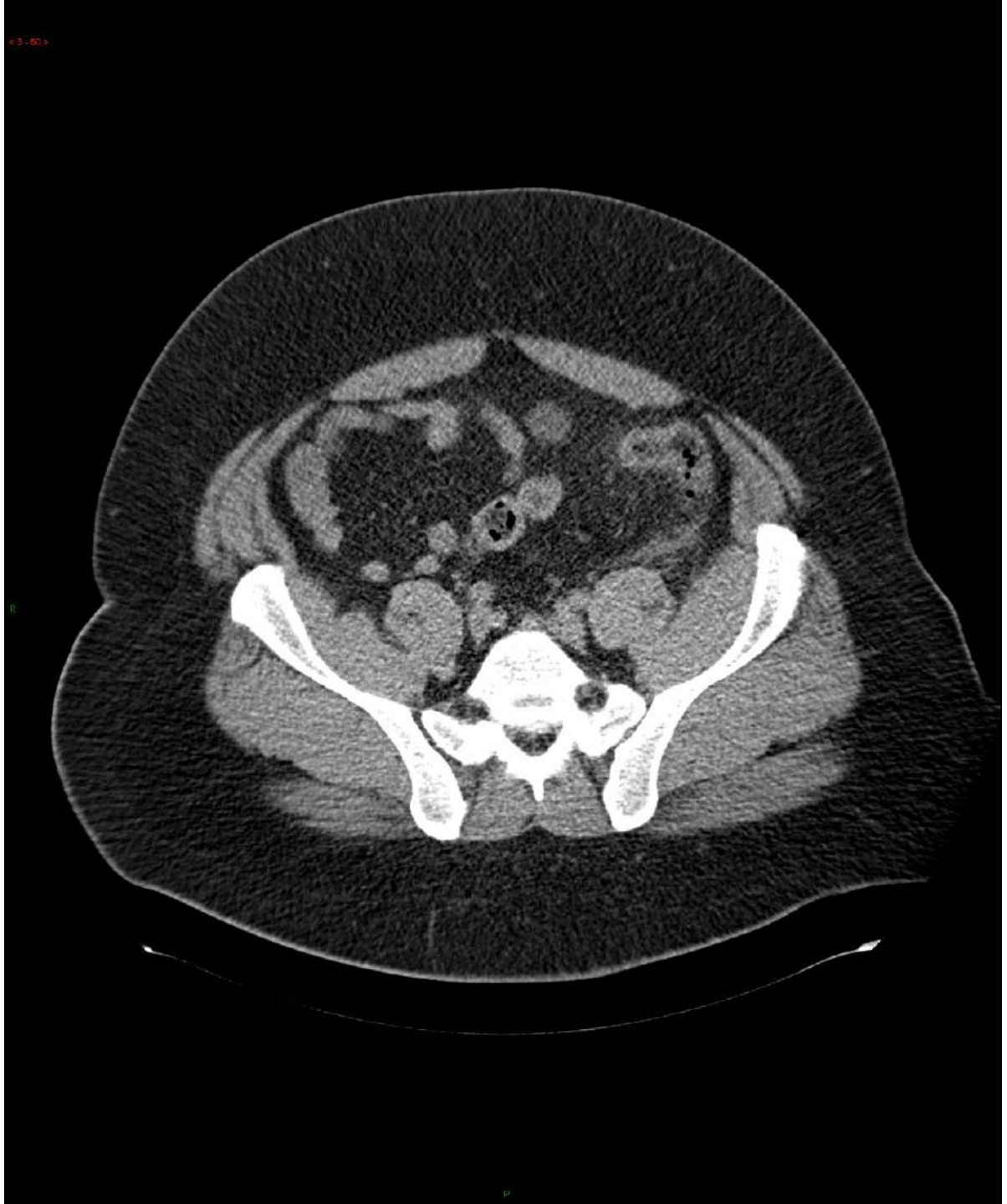
Andrew Ng

# 3D data



Andrew Ng

# 3D data



Andrew Ng

# 3D data

Andrew Ng

# 3D data

Andrew Ng

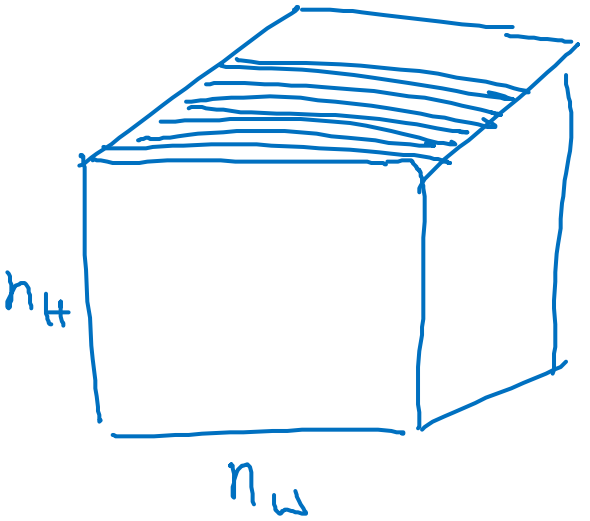
# 3D data

Andrew Ng

# 3D data

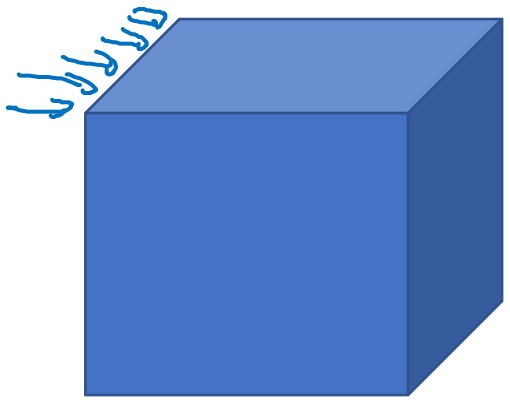
Andrew Ng

# 3D data



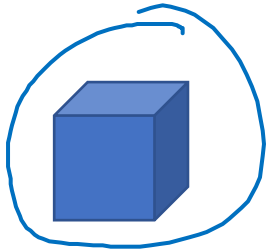

Andrew Ng

# 3D convolution



3D volume

*



3D filter

$14 \times 14 \times 14 \times 1$ ($n_c$)

$* \; 5 \times 5 \times 5 \times 1$

16 filters.

$\rightarrow 10 \times 10 \times 10 \times 16$

$* \; 5 \times 5 \times 5 \times 16$

32 filters

$\rightarrow 6 \times 6 \times 6 \times 32$