deeplearning.ai
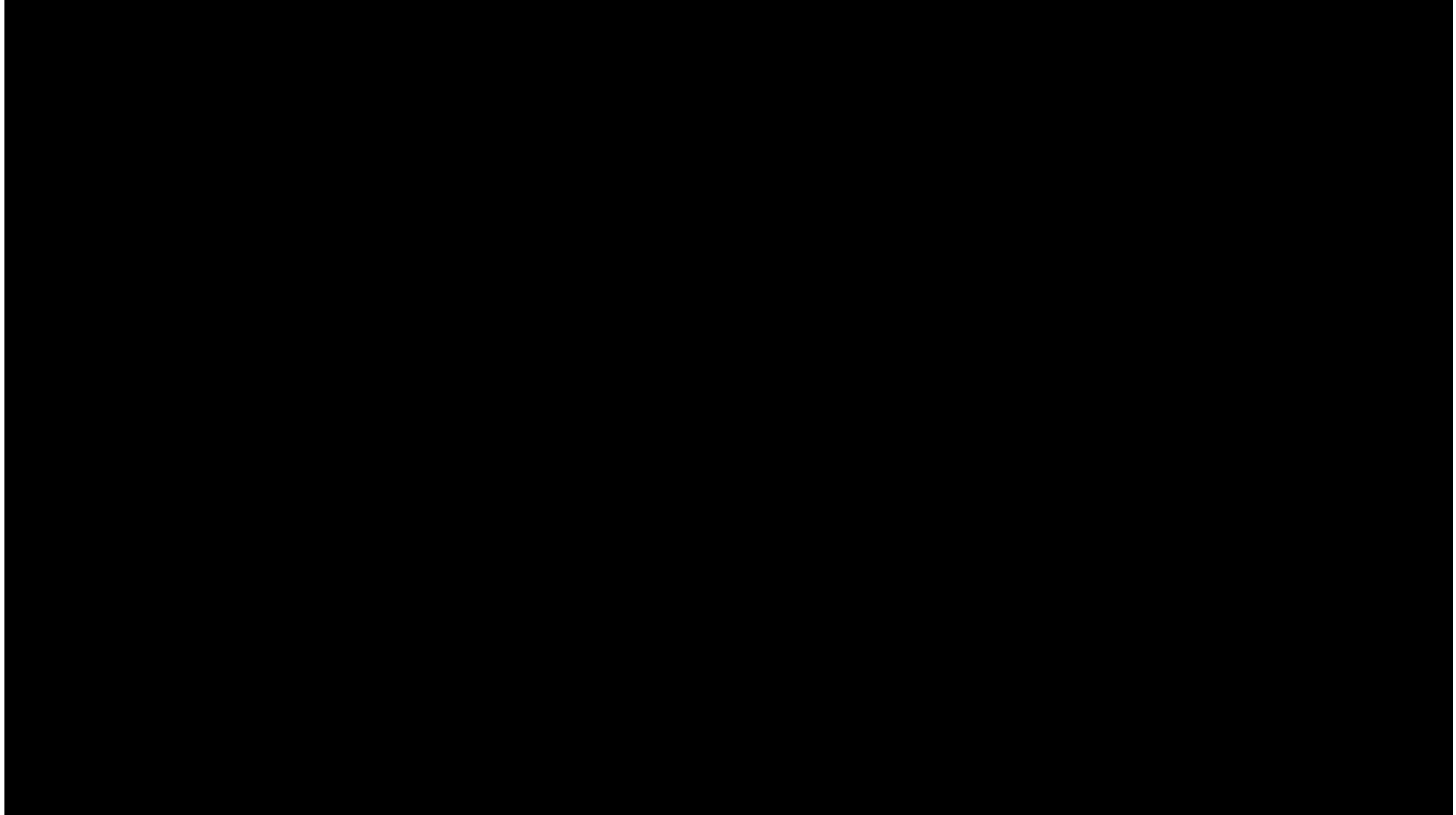
Face recognition

What is face recognition?

Face recognition

Andrew Ng

Face verification vs. face recognition

→ Verification   1:1   99%

- Input image, name/ID
- Output whether the input image is that of the   99.9
  claimed person

→ Recognition   1:K

- Has a database of K persons
- Get an input image   K=100 ←
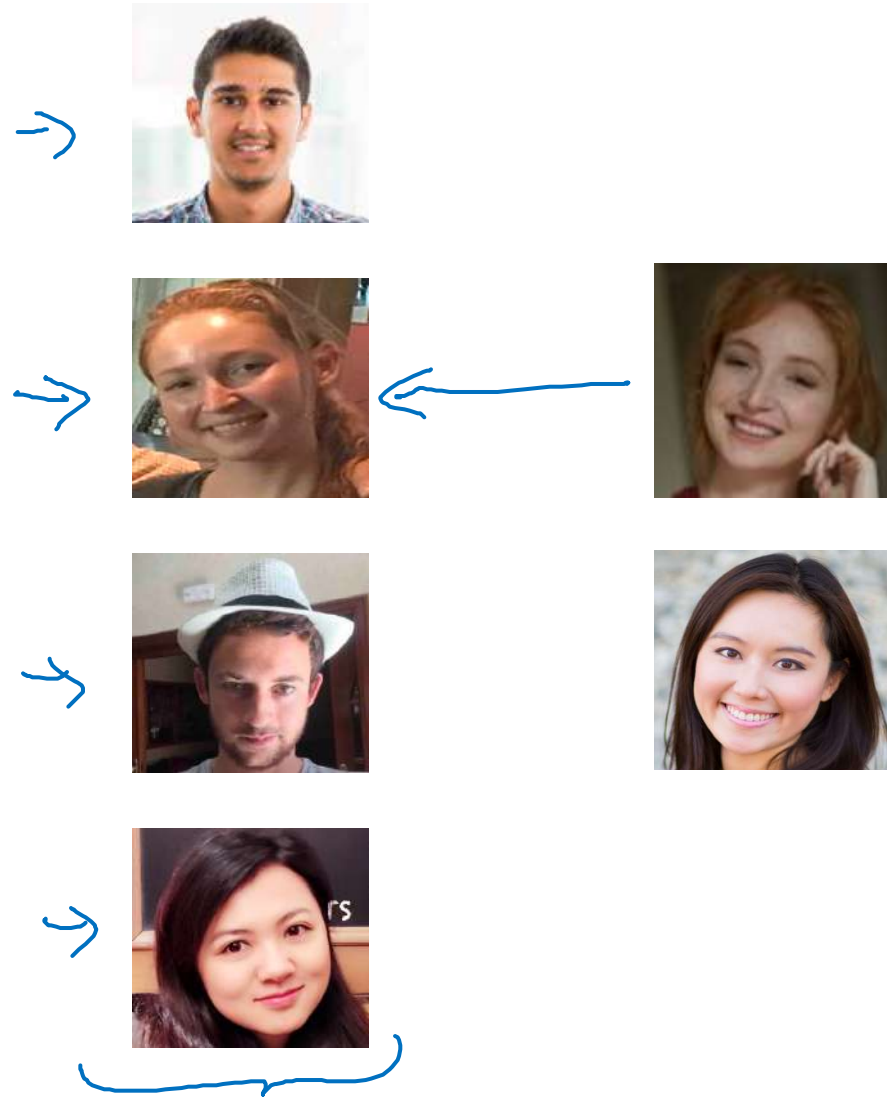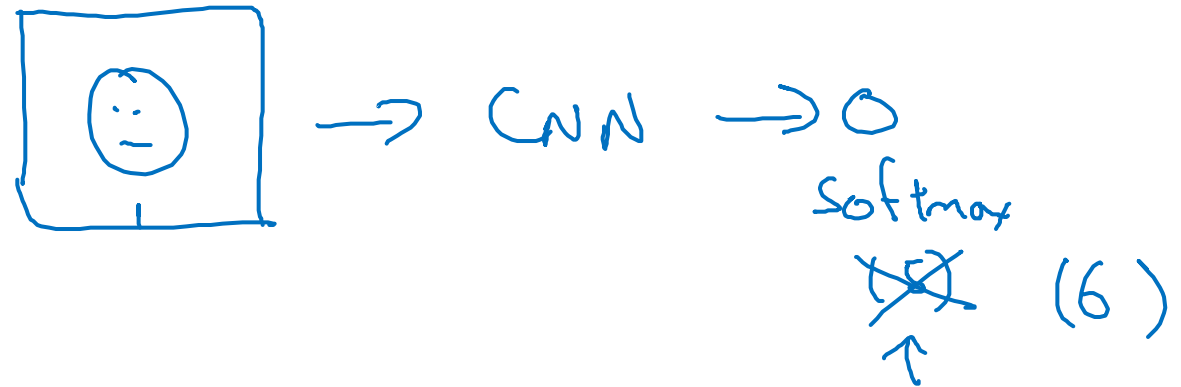- Output ID if the image is any of the K persons (or
  "not recognized")

Face recognition

One-shot learning

deeplearning.ai

One-shot learning

Learning from one example to recognize the person again

Andrew Ng

Learning a "similarity" function
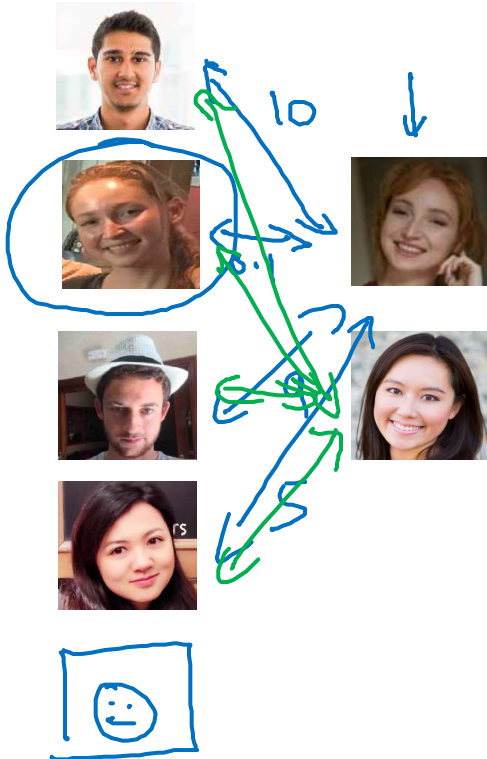
→ d(img1,img2) = degree of difference between images

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

              $> \tau$ "different"   } Verification.

10

$d(img1, img2)$

Andrew Ng

Face recognition
_____

Siamese network

deeplearning.ai

Siamese network

128

$x^{(1)}$

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

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

128

$x^{(2)}$

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

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

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

Andrew Ng

Goal of learning

$$f(x^{(1)})$$
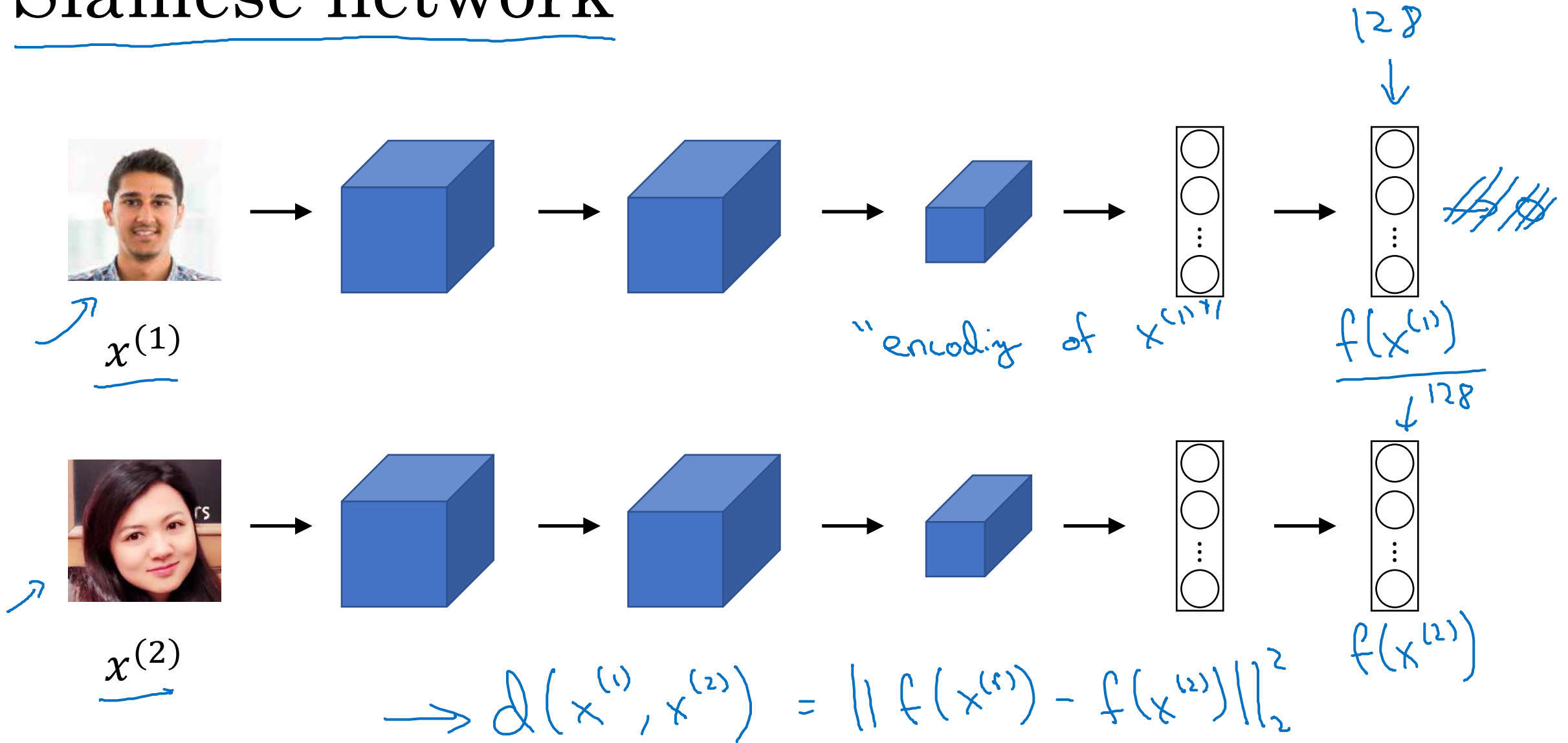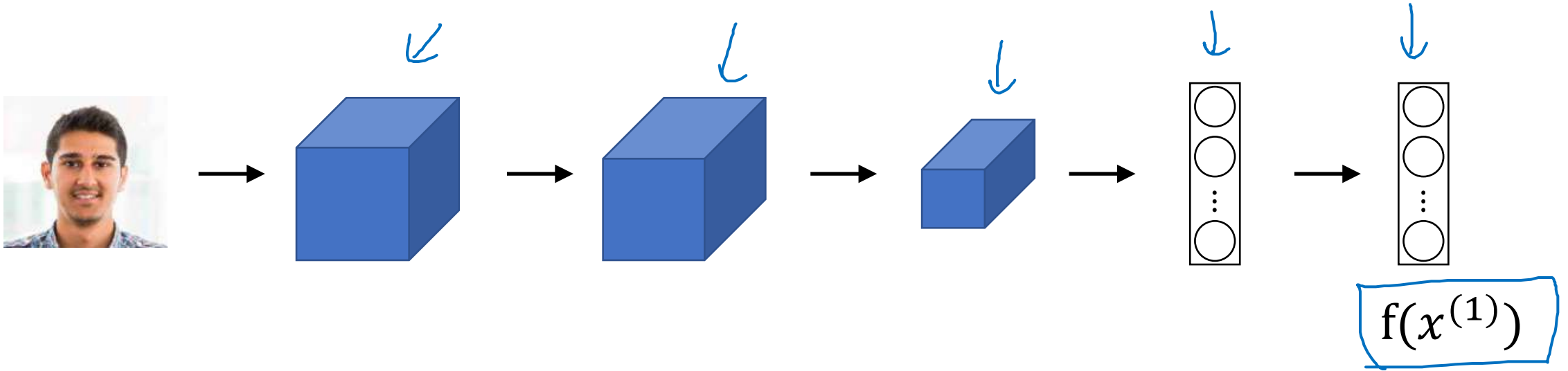
Parameters of NN define an encoding $f(x^{(i)})$ — 128

Learn parameters so that:

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

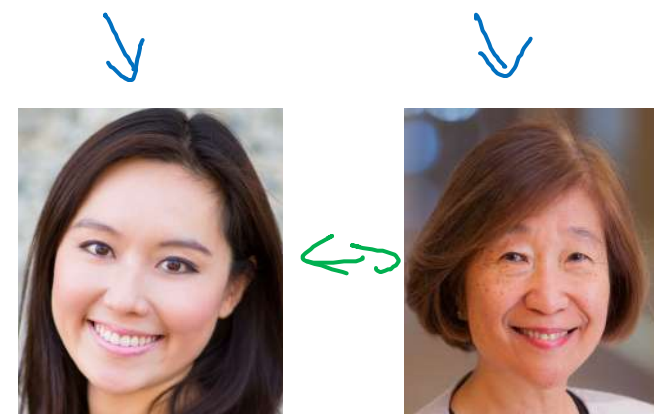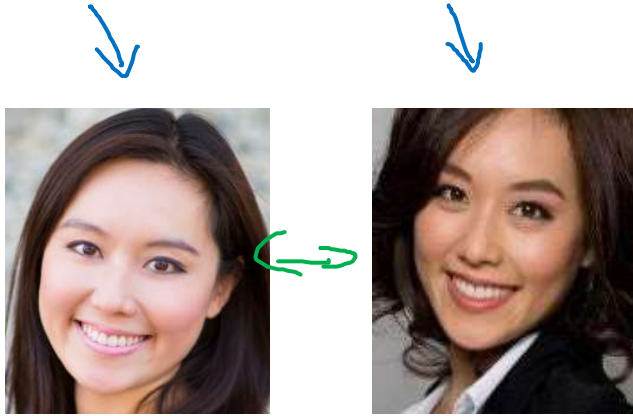If $x^{(i)}, x^{(j)}$ are different persons, $\left\|f(x^{(i)}) - f(x^{(j)})\right\|^2$ is large.

Andrew Ng

Face recognition

Triplet loss

deeplearning.ai

Learning Objective

Anchor   Positive   Anchor   Negative

A $d(A,P) = 0.5$ > 0.2 A $d(A,N) = 0.5 \quad 0.7$

Want:

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

$$\boxed{\underbrace{\| f(A) - f(P) \|^2}_{0} - \underbrace{\| f(A) - f(N) \|^2}_{0} + \underbrace{\alpha}_{margin} \leq 0} \quad \text{\#/\#} \qquad f(img) = \vec{0}$$

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

Andrew Ng

Loss function

Given 3 images A, P, N:

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

$\leq 0 \quad > 0$

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

A, P

Training set: 10k pictures of 1k persons

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

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

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

Andrew Ng

Training set using triplet loss

Anchor   Positive   Negative

$$J$$

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

Face recognition

Face verification and binary classification

deeplearning.ai

Learning the similarity function

new

$x^{(i)}$

$x^{(j)}$

Database $\times$

$\rightarrow$ f($x^{(i)}$)

$\rightarrow$ f($x^{(j)}$)

k

$\hat{y}$

1

0

Precompute

$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_k \left| f(x^{(i)})_k - f(x^{(j)})_k \right| + b \right)$$

$$\frac{\left(f(x^{(i)})_k - f(x^{(j)})_k\right)^2}{f(x^{(i)})_k + f(x^{(j)})_k}$$

χ^2

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

Andrew Ng
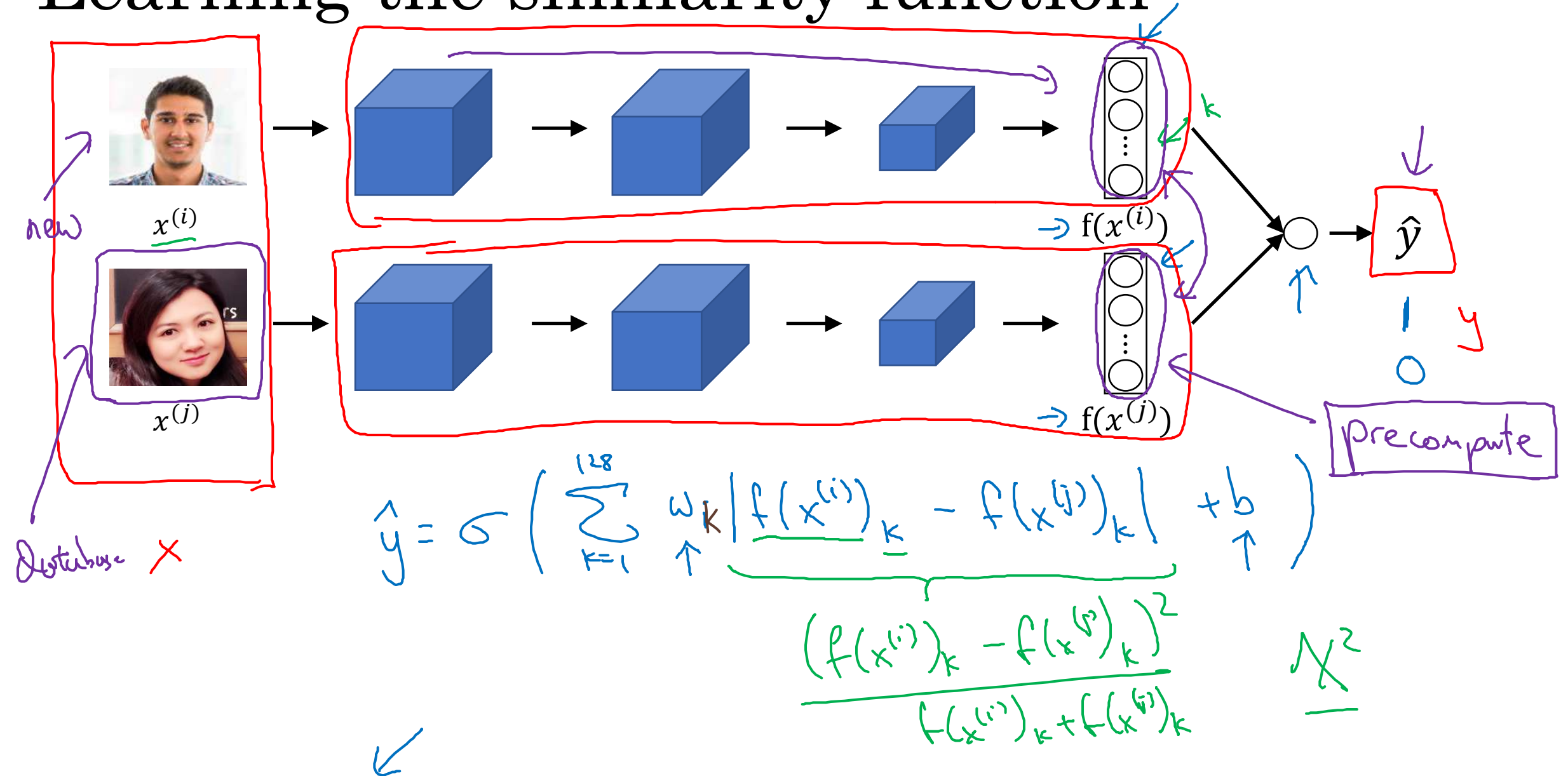
Face verification supervised learning

x y

1   "Same"

0   "different"

0

1

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

Andrew Ng

Neural Style Transfer

What is neural style transfer?

deeplearning.ai

Neural style transfer

Content (C) Style (S)

Content (C) Style (S)

Generated image (G)

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

224×224×3　　110×110×96　　55×55×96　　26×26×256　13×13×256　13×13×384　13×13×384　6×6×256　FC 4096　FC 4096　$\hat{y}$
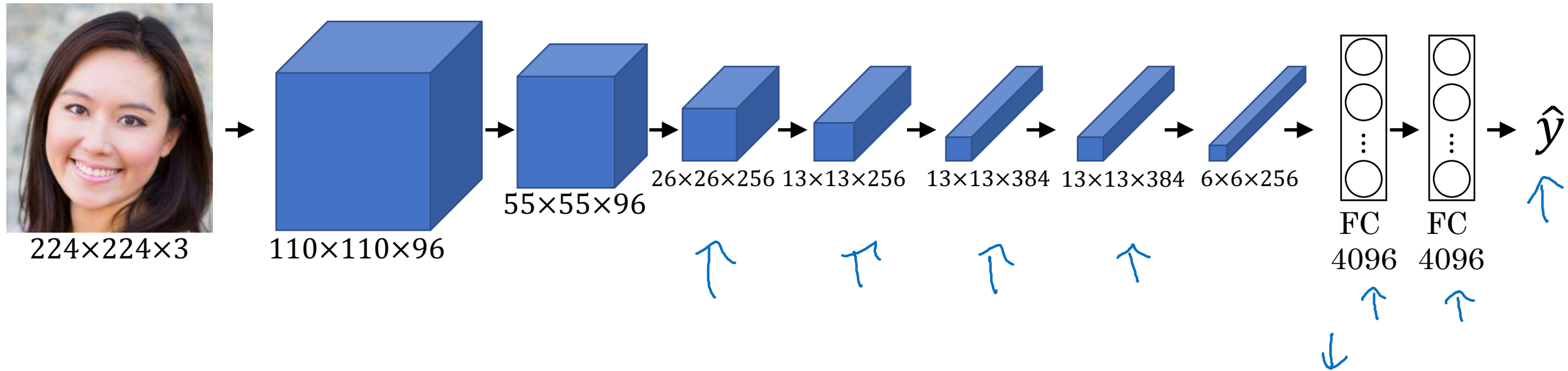
Pick a unit in layer 1. Find the nine image patches that maximize the unit's activation.

Repeat for other units.

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

Andrew Ng
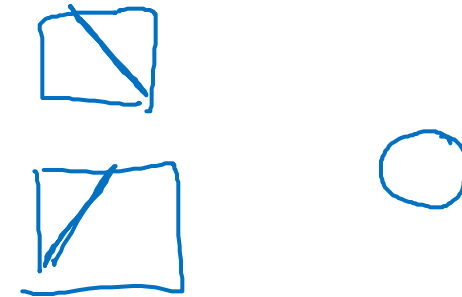
Visualizing deep layers
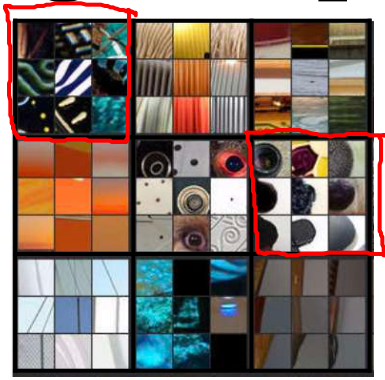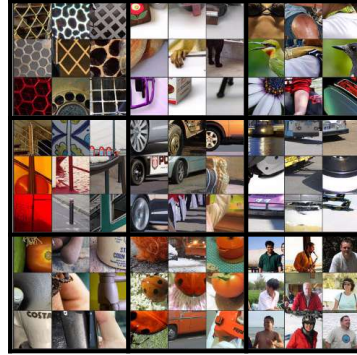
Layer 1   Layer 2   Layer 3   Layer 4   Layer 5
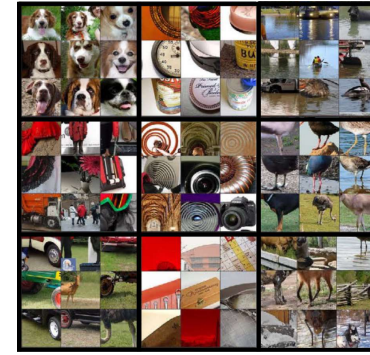
Visualizing deep layers: Layer 1
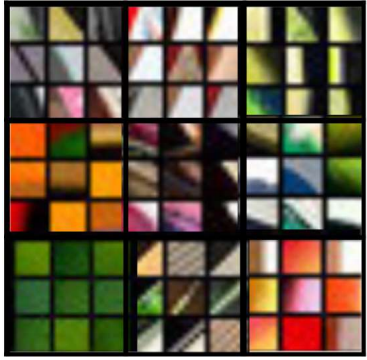
Layer 1   Layer 2   Layer 3   Layer 4   Layer 5
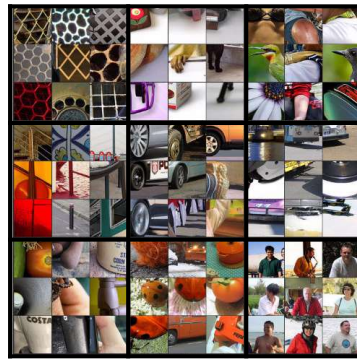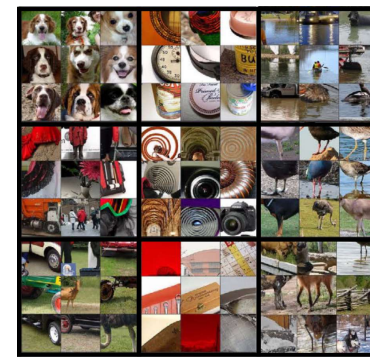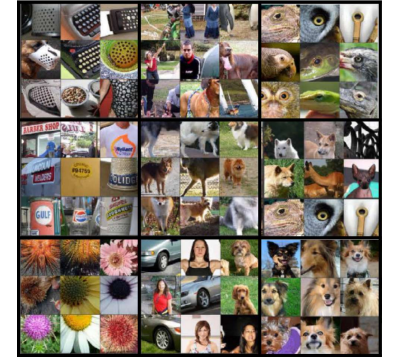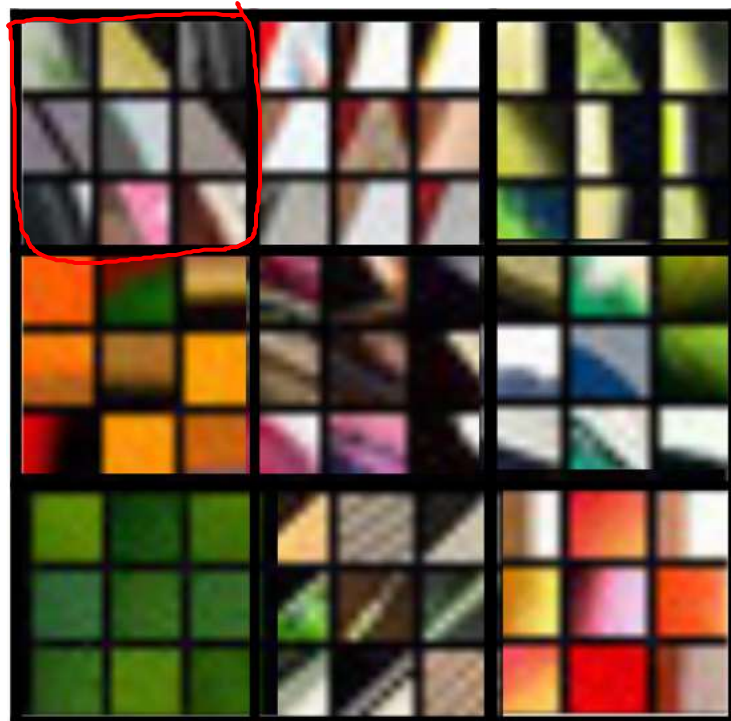
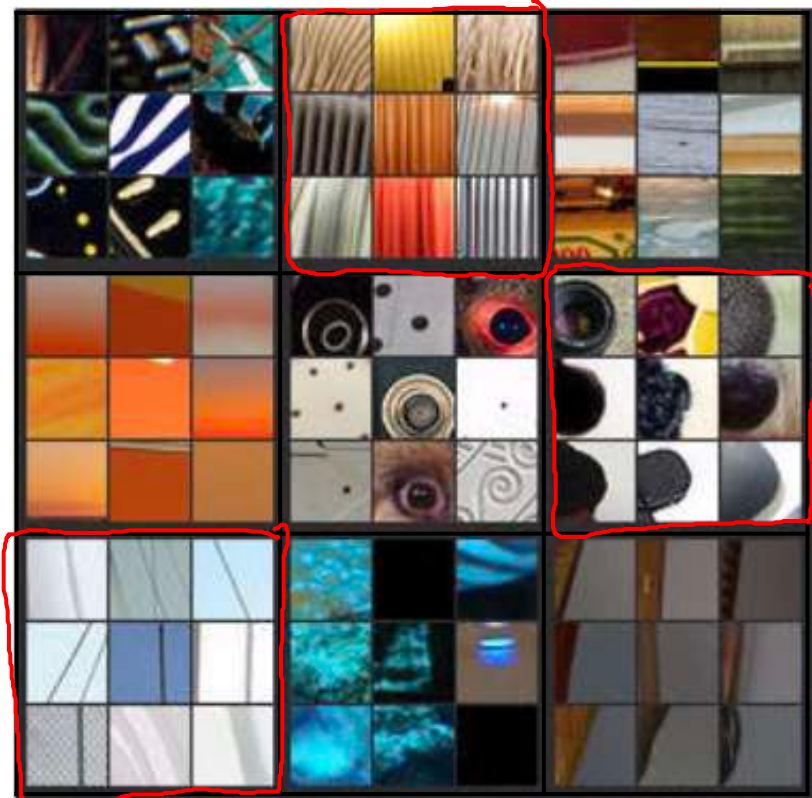# Visualizing deep layers: Layer 2

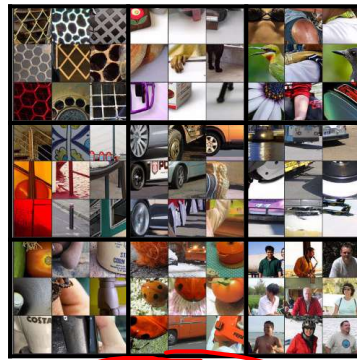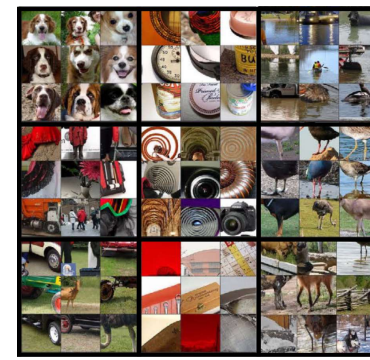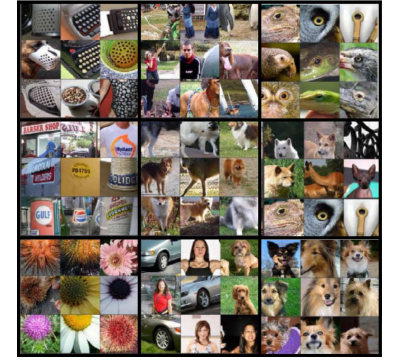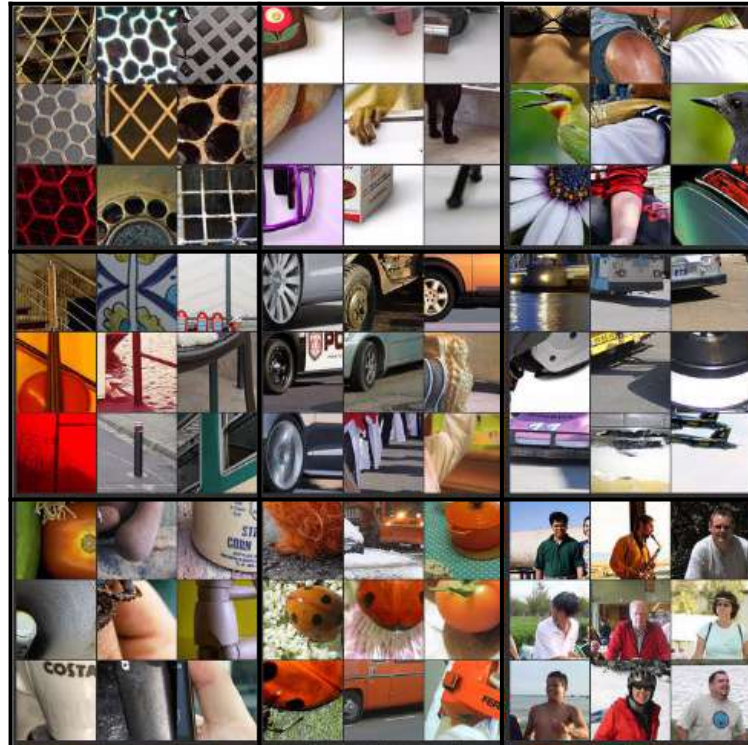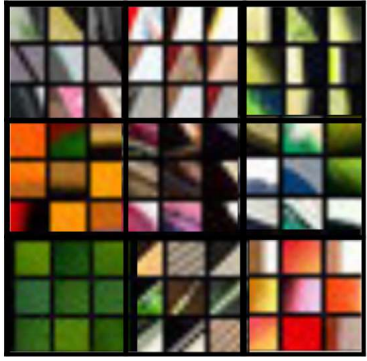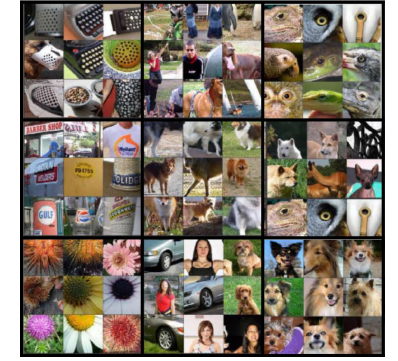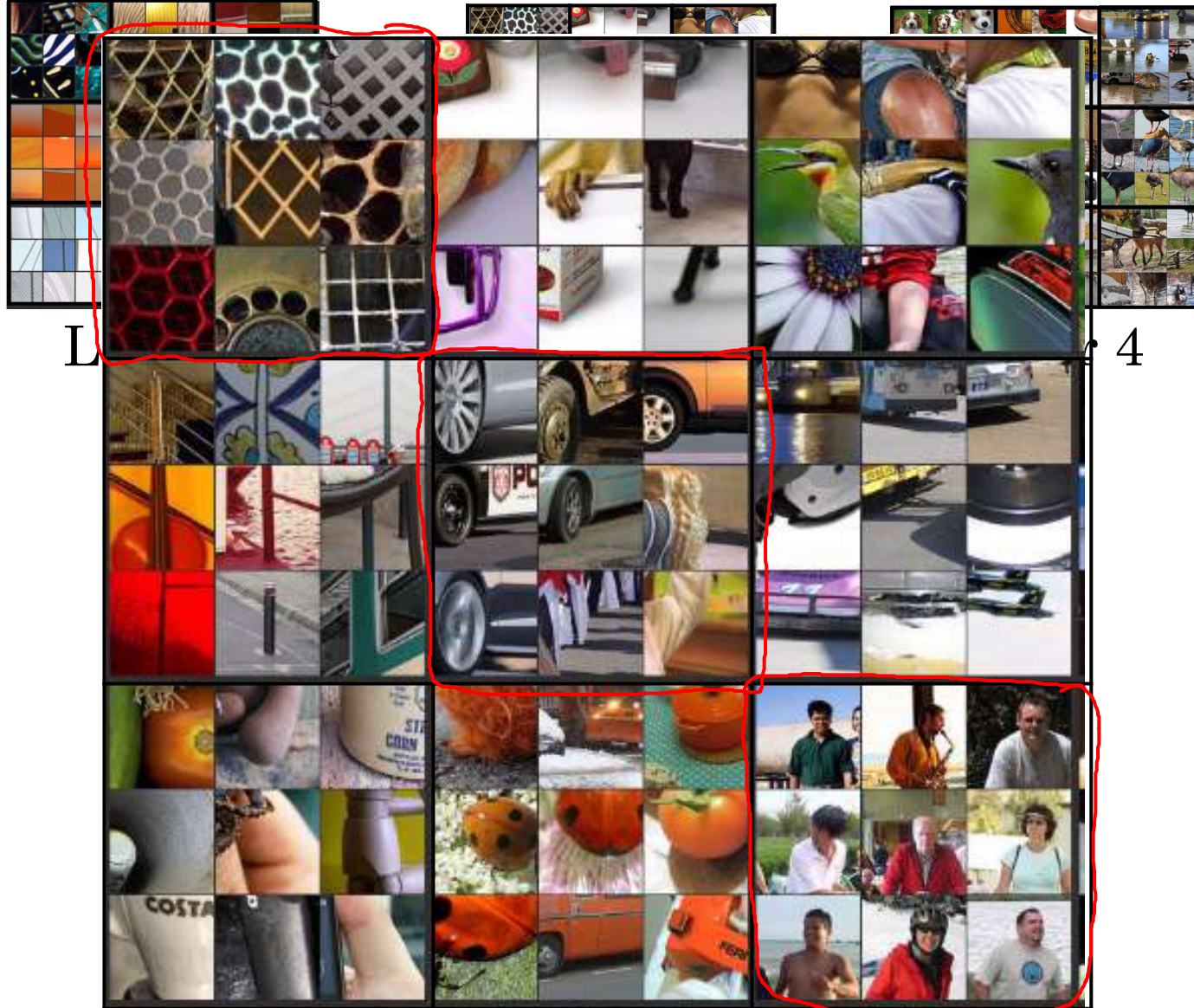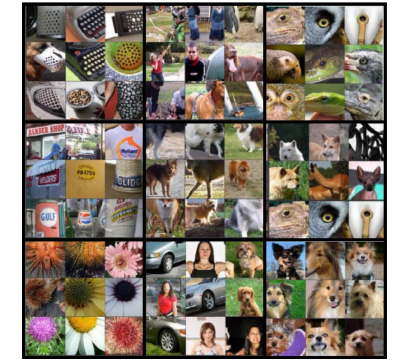Layer 1   Layer 2   Layer 3   Layer 4   Layer 5
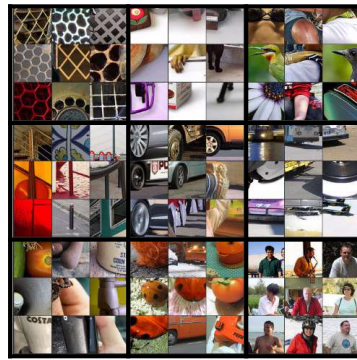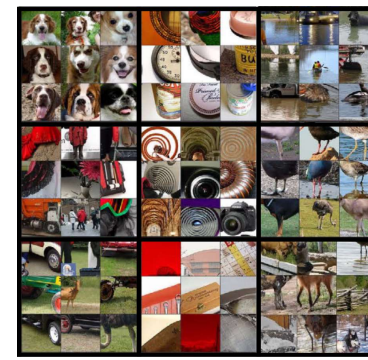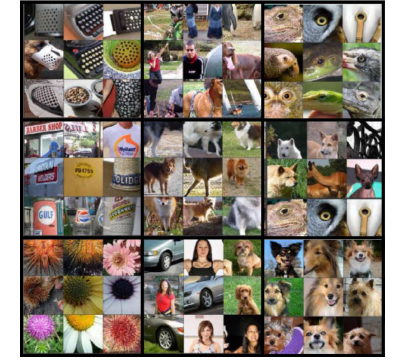
Visualizing deep layers: Layer 3

Layer 1   Layer 2   Layer 3   Layer 4   Layer 5

Andrew Ng

Visualizing deep layers: Layer 3

Layer 1   Layer 2   Layer 4   Layer 5

Andrew Ng

Visualizing deep layers: Layer 4
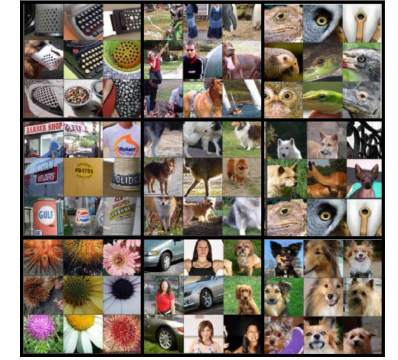

Layer 4

Layer 5

Andrew Ng
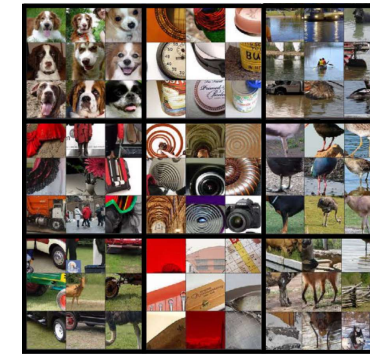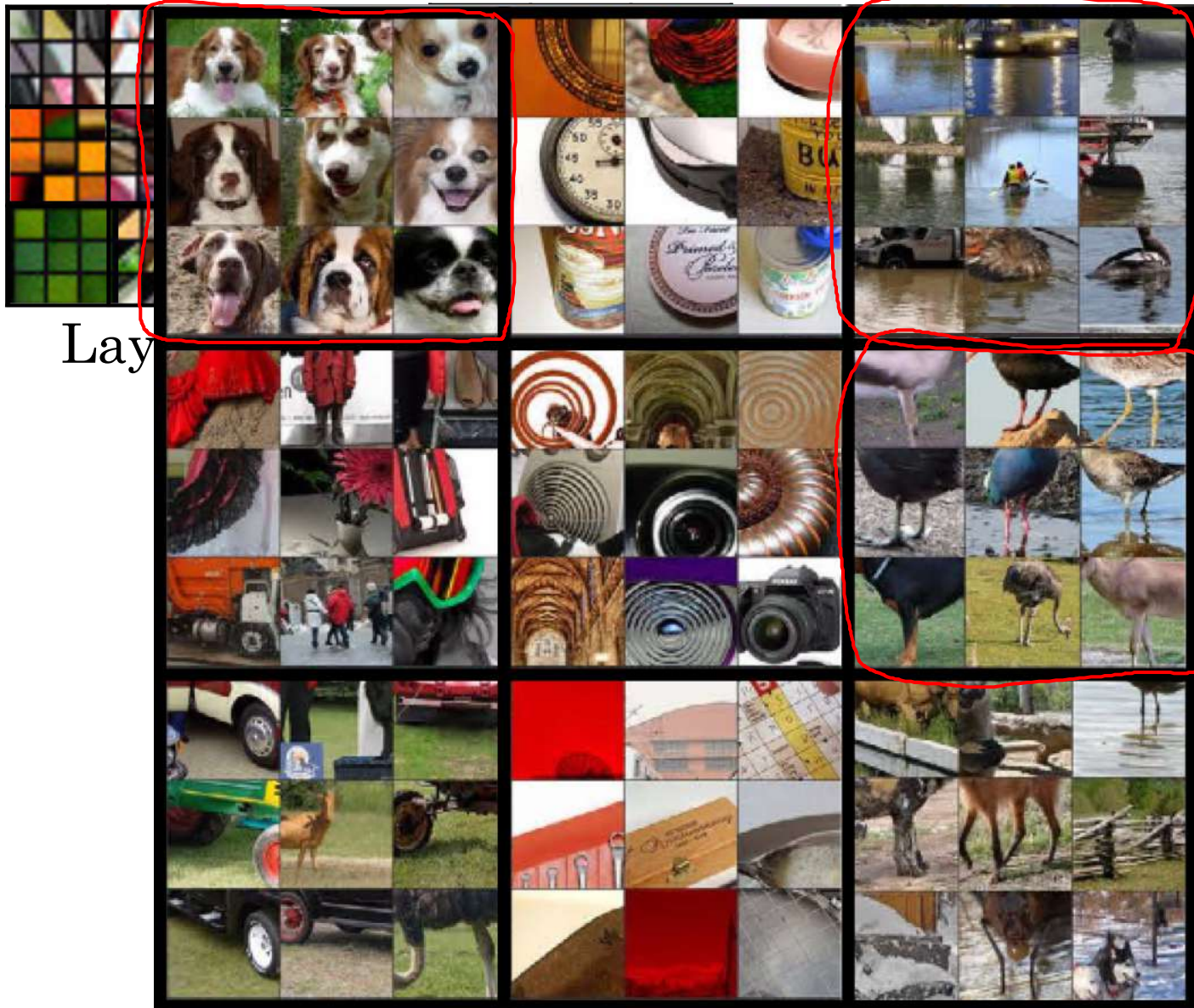
Visualizing deep layers: Layer 5

Layer 1
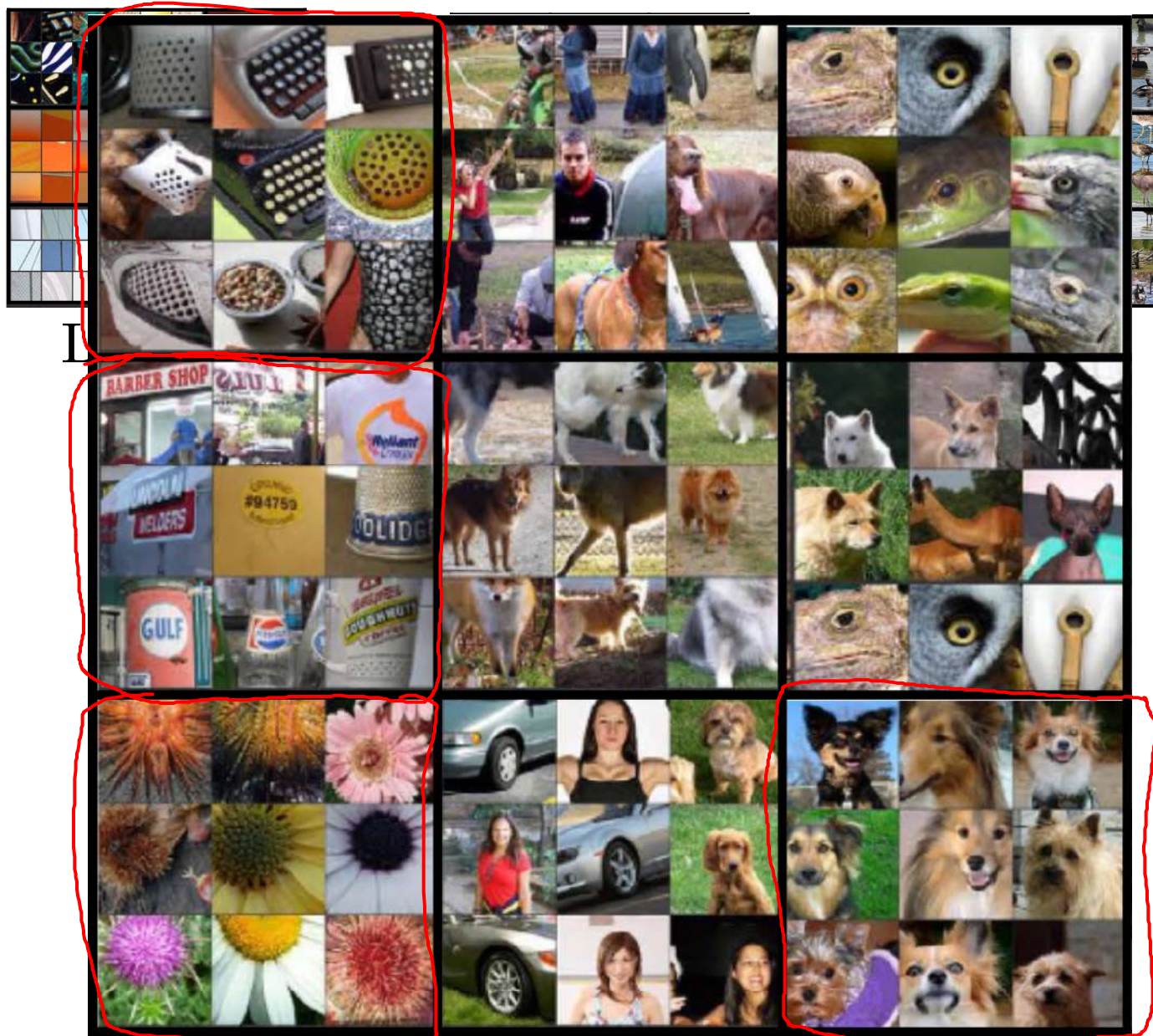
Layer 5

Andrew Ng

Neural Style Transfer

Cost function

deeplearning.ai

Neural style transfer cost function

Content C

Style S

Generated image G

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

Andrew Ng

Find the generated image G

1. Initiate G randomly

   G: 100×100×3

   $\uparrow$
   RGB

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

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

Andrew Ng

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

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

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

Neural Style Transfer

Style cost function

deeplearning.ai
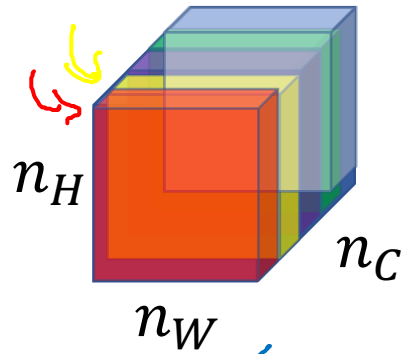
Meaning of the "style" of an image
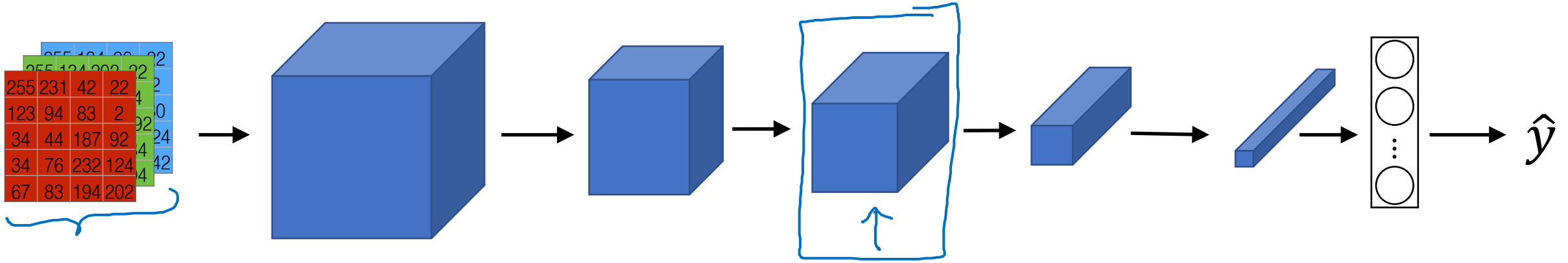
Say you are using layer l's activation to measure "style."
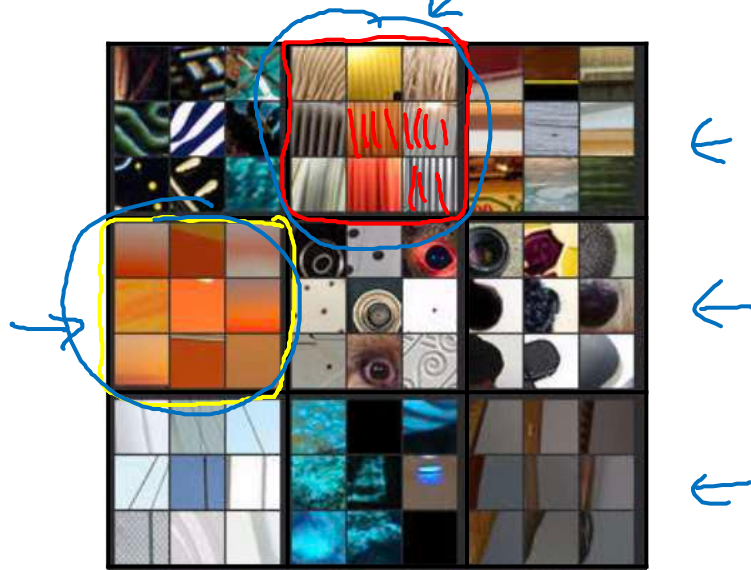Define style as correlation between activations across channels.
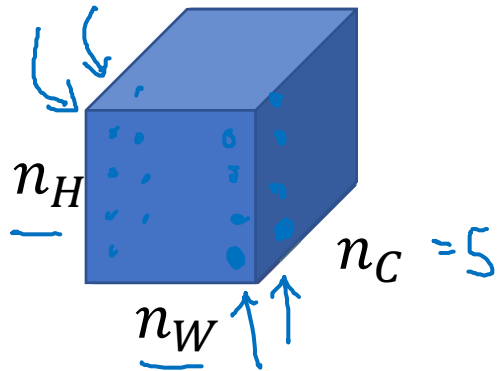
How correlated are the activations across different channels?

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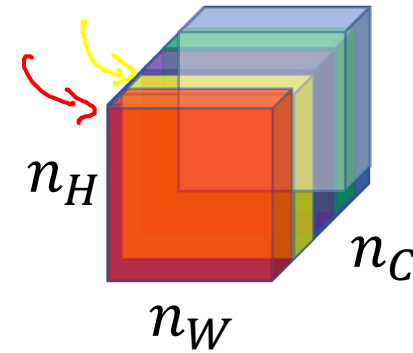
Andrew Ng

Intuition about style of an image

Style image

Generated Image

Correlated?

Uncorrelated

n_H n_C n_W

n_H n_C n_W

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

Andrew Ng

Style matrix

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

H W C

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"

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

β

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

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

Style cost function

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

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

$$J_{style}(S,G) = \sum_l \lambda^{[l]} J_{style}^{[l]}(S,G)$$
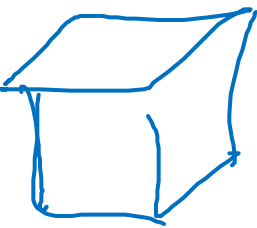
$$J(G) = \alpha J_{content}(C,G) + \beta J_{style}(S,G)$$

$$G$$

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

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

2D input image
14×14

$14 \times 14 \times \underline{3} \quad * \quad 5 \times 5 \times 3$

$\rightarrow 10 \times 10 \times 16$

$10 \times 10 \times \underline{16} \quad * \quad 5 \times 5 \times \underline{16}$

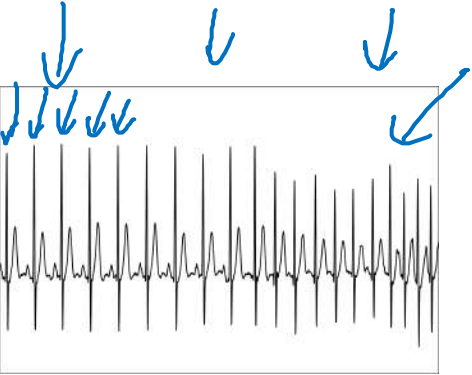$\rightarrow 6 \times 6 \times \underline{32}$

$14 \times \underline{1} \quad * \quad 5 \times \underline{1}$

$\rightarrow 10 \times \underline{16}$

$10 \times 16 \quad * \quad 5 \times \underline{16}$

$\rightarrow 6 \times 32$
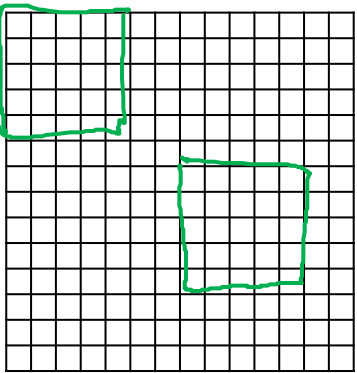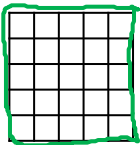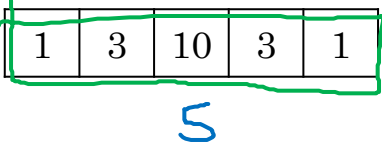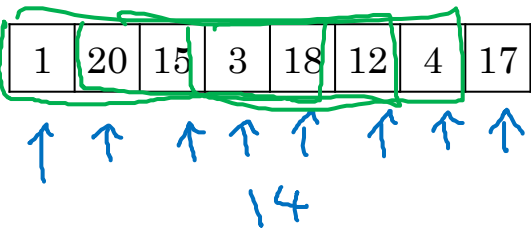
1	20	15	3	18	12	4	17

14

*

1	3	10	3	1

5

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

n_H

n_W

Andrew Ng
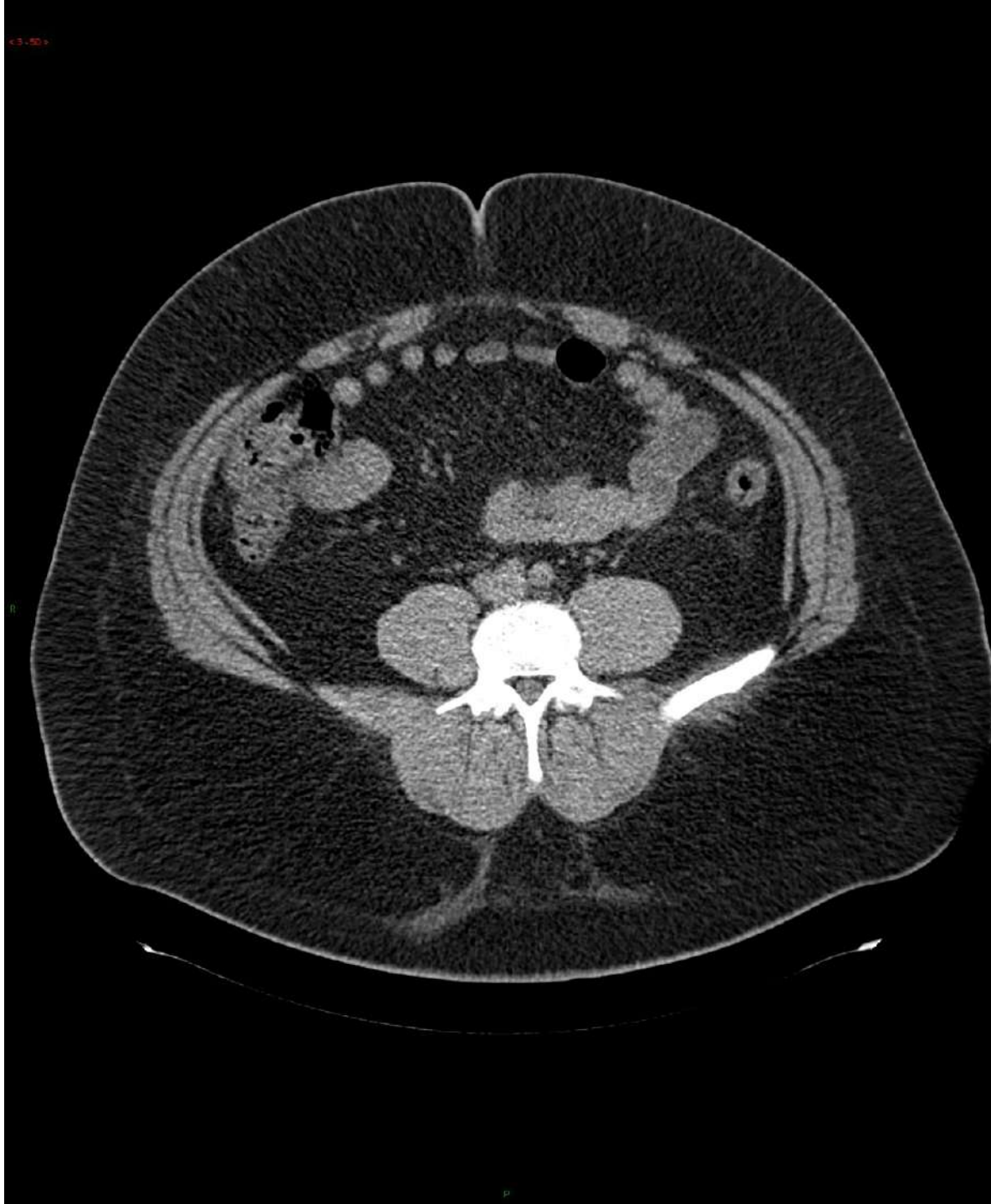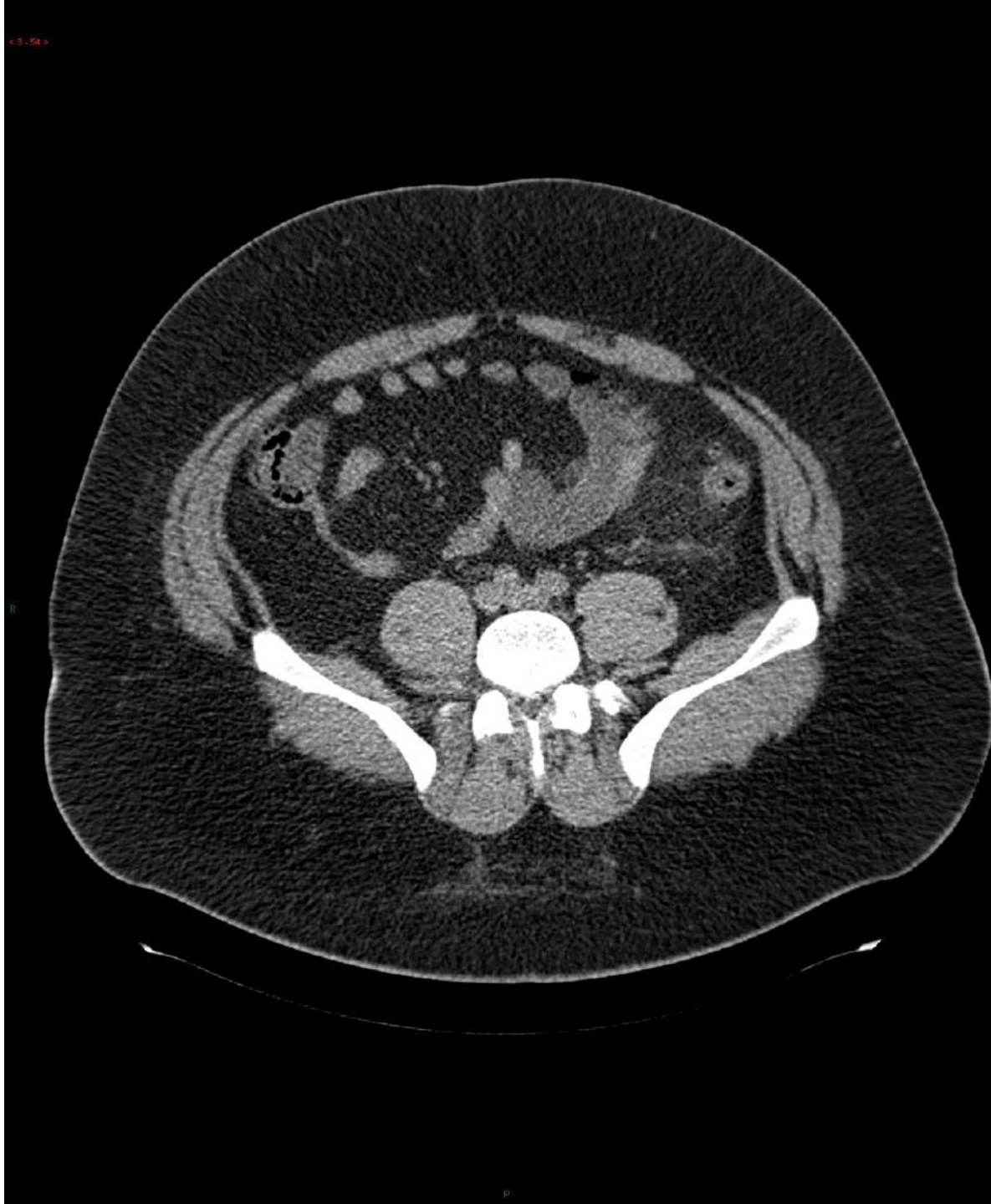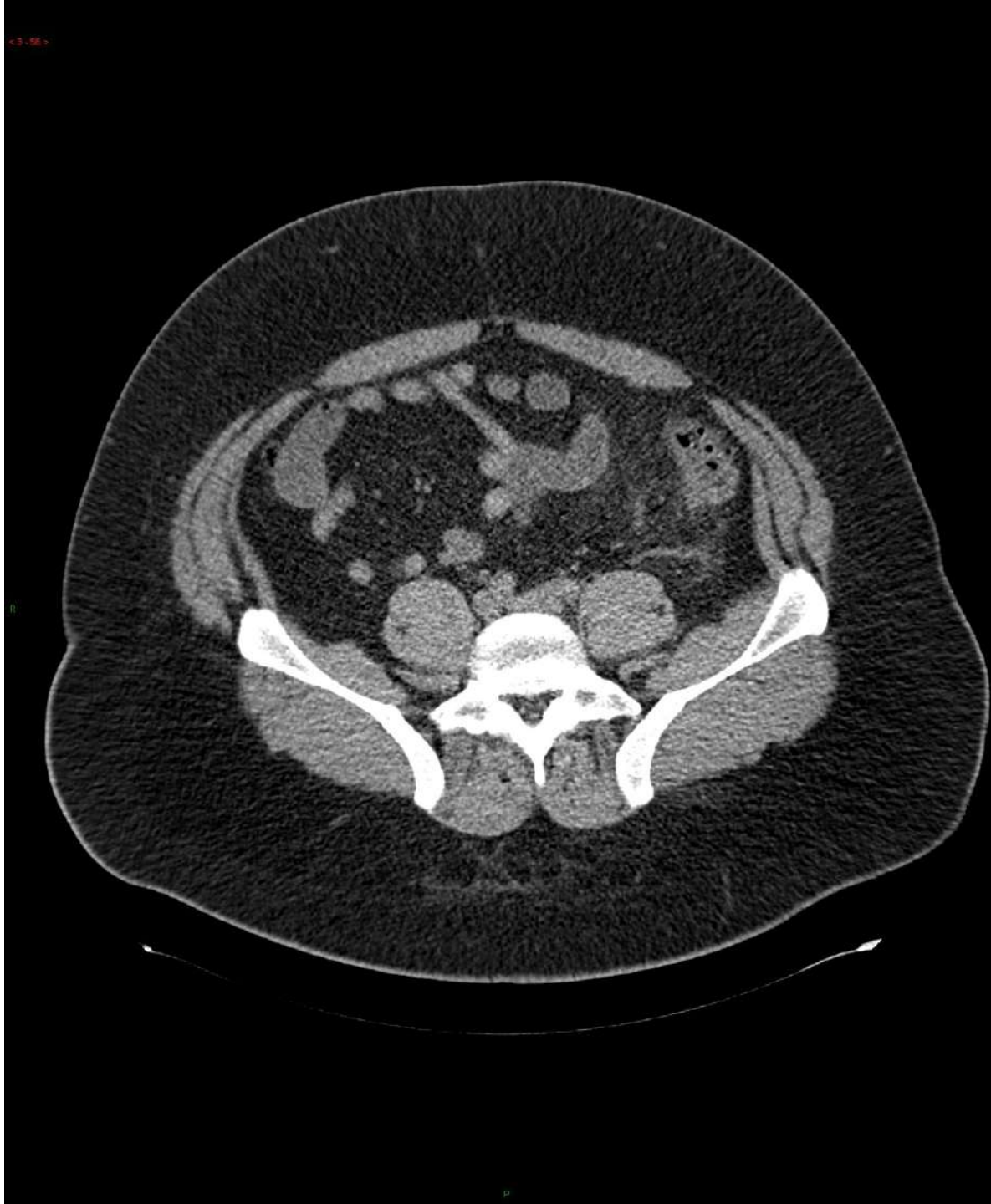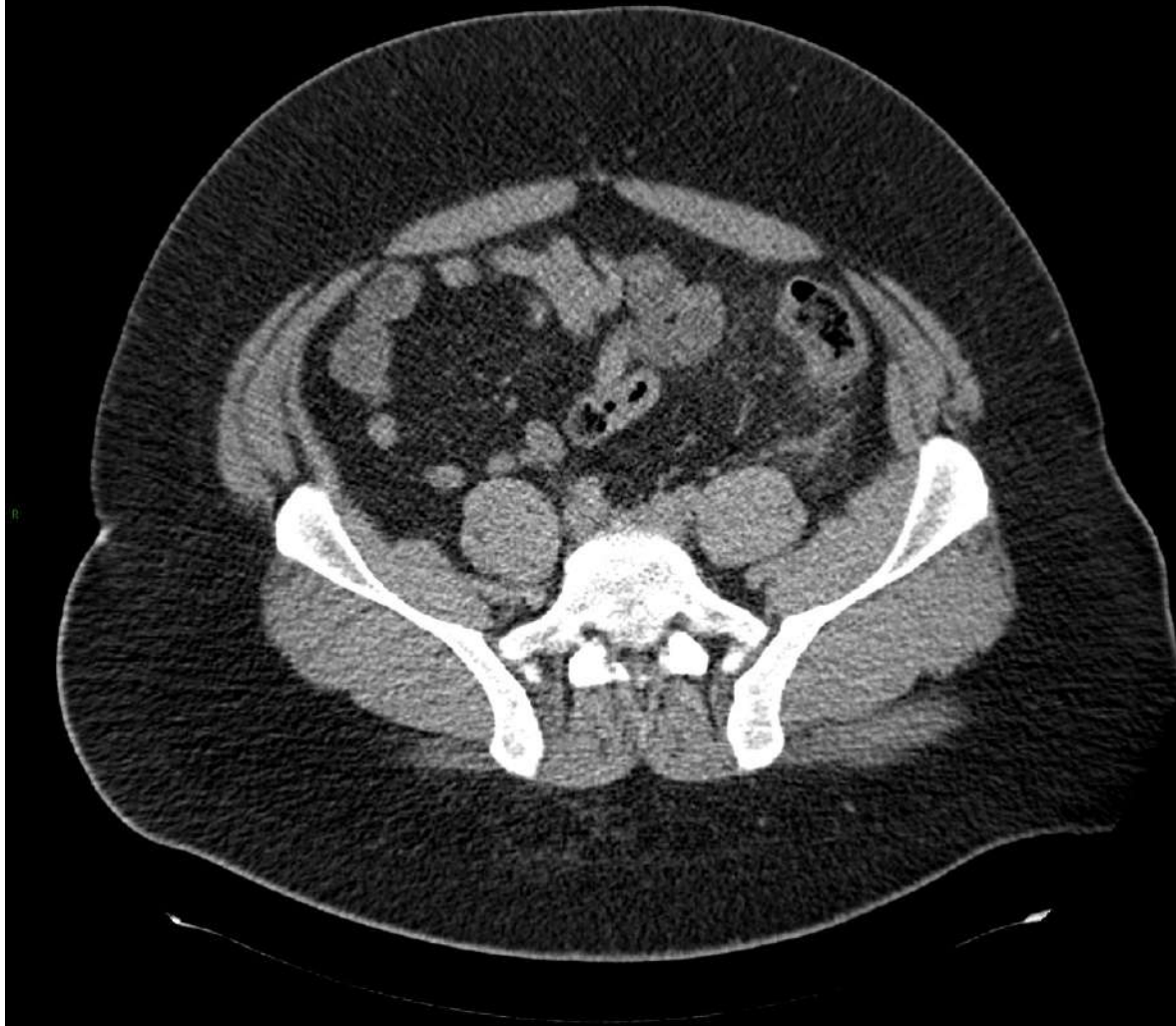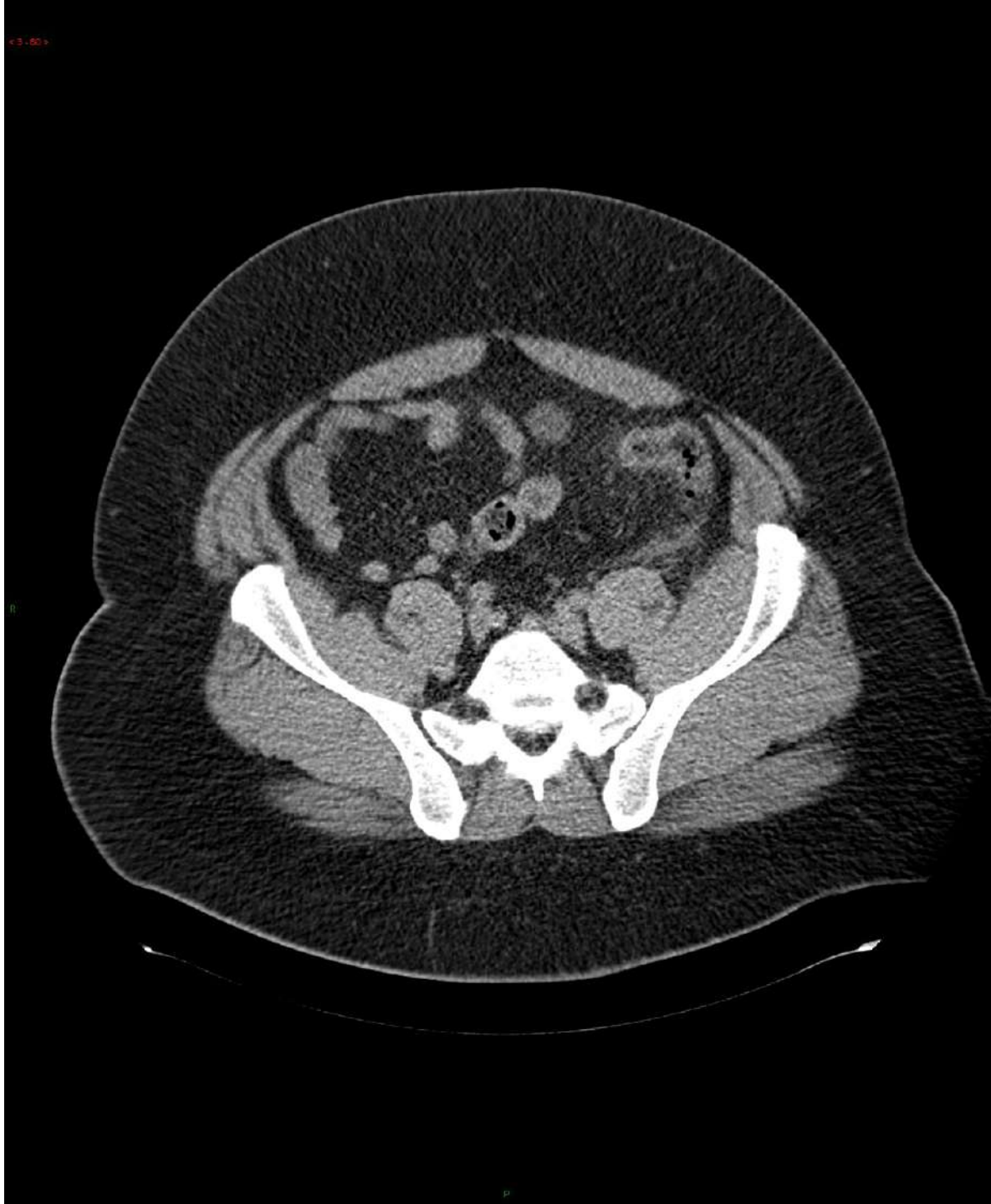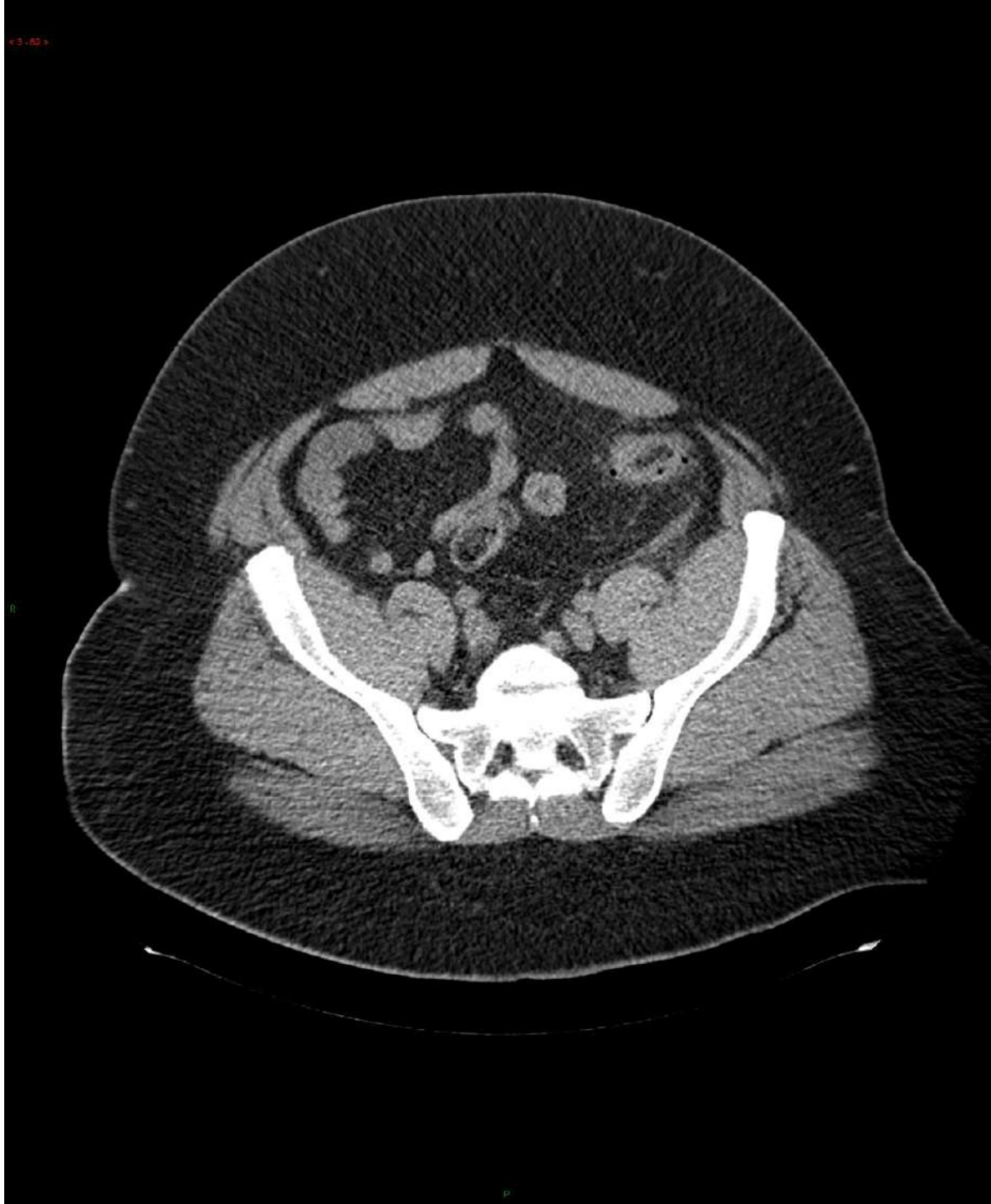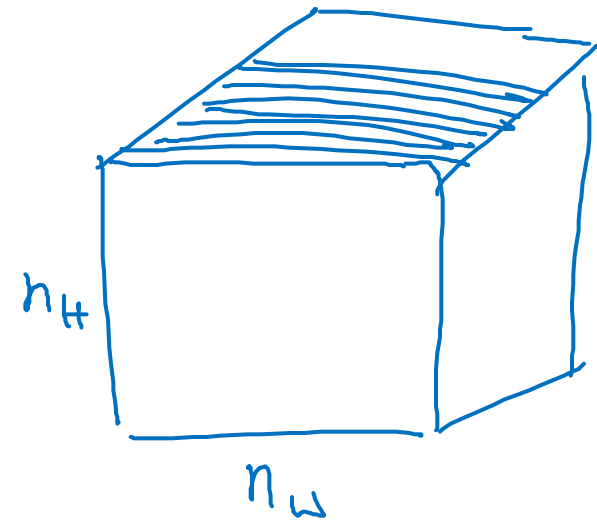
3D convolution

3D volume * 3D filter

$$14 \times 14 \times 14 \times 1 \quad n_c$$

$$* \; 5 \times 5 \times 5 \times 1 \qquad \text{16 filte.}$$

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

$$* \; 5 \times 5 \times 5 \times 16 \qquad \text{32 filters}$$

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