## Copyright Notice

These slides are distributed under the Creative Commons License.

<u>DeepLearning.Al</u> makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite <u>DeepLearning.Al</u> as the source of the slides.

For the rest of the details of the license, see <a href="https://creativecommons.org/licenses/by-sa/2.0/legalcode">https://creativecommons.org/licenses/by-sa/2.0/legalcode</a>



## 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
- → 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")

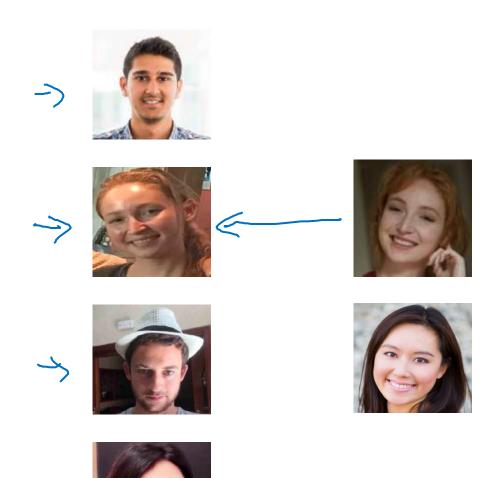




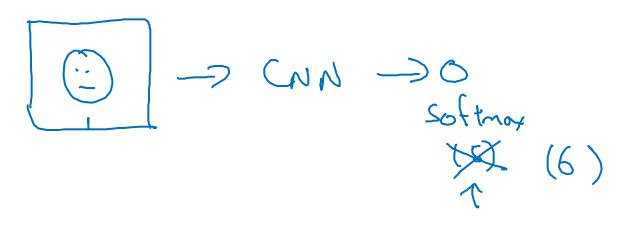
## Face recognition

## One-shot learning

#### One-shot learning



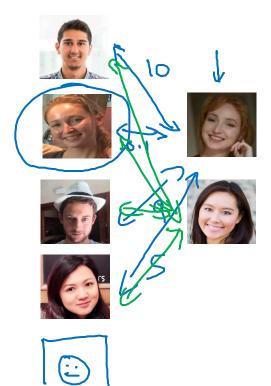
Learning from one example to recognize the person again



## Learning a "similarity" function

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

If 
$$d(img1,img2) \leq \tau$$
 "some"  $> \tau$  "Quiterest"



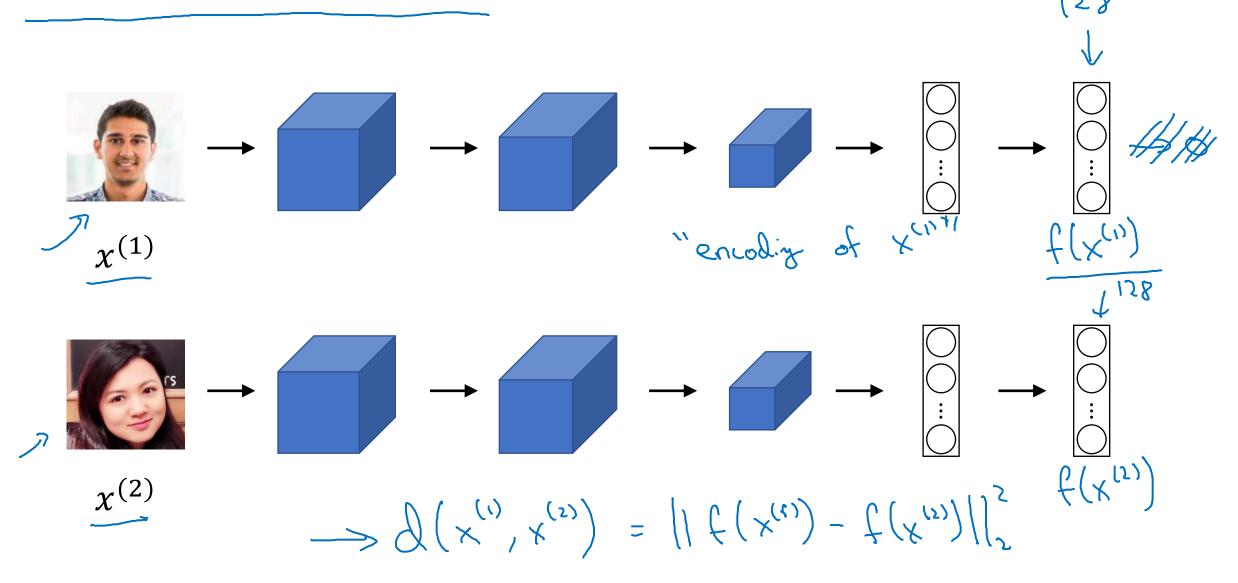




## Face recognition

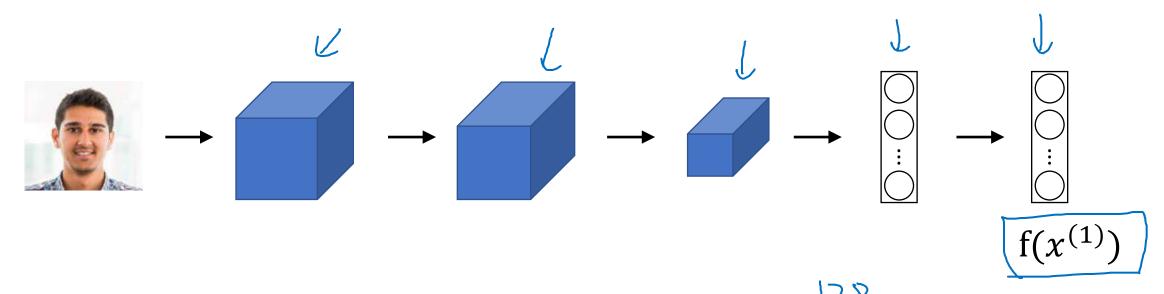
#### Siamese network

#### Siamese network





#### Goal of learning



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

Learn parameters so that:

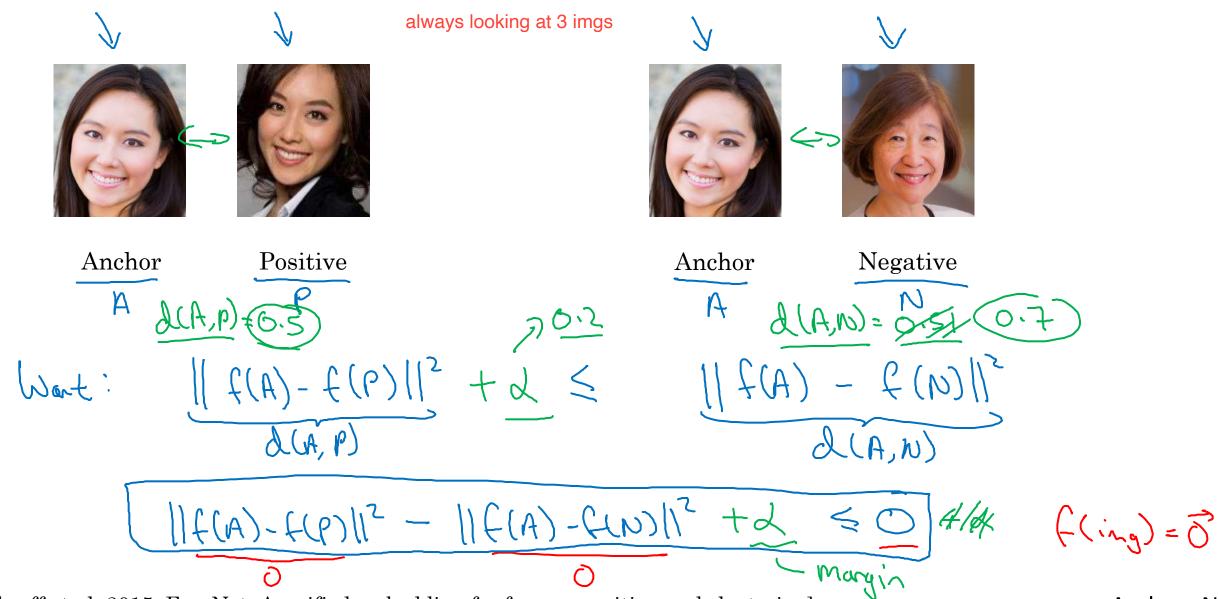
If 
$$x^{(i)}$$
,  $x^{(j)}$  are the same person,  $\|f(x^{(i)}) - f(x^{(j)})\|^2$  is small.  
If  $x^{(i)}$ ,  $x^{(j)}$  are different persons,  $\|f(x^{(i)}) - f(x^{(j)})\|^2$  is large.



## Face recognition

## Triplet loss

#### Learning Objective



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

avoid trival result

Andrew Ng

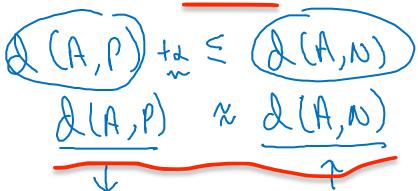
Loss function

Training set: 10k pictures of 1k persons

## Choosing the triplets A,P,N

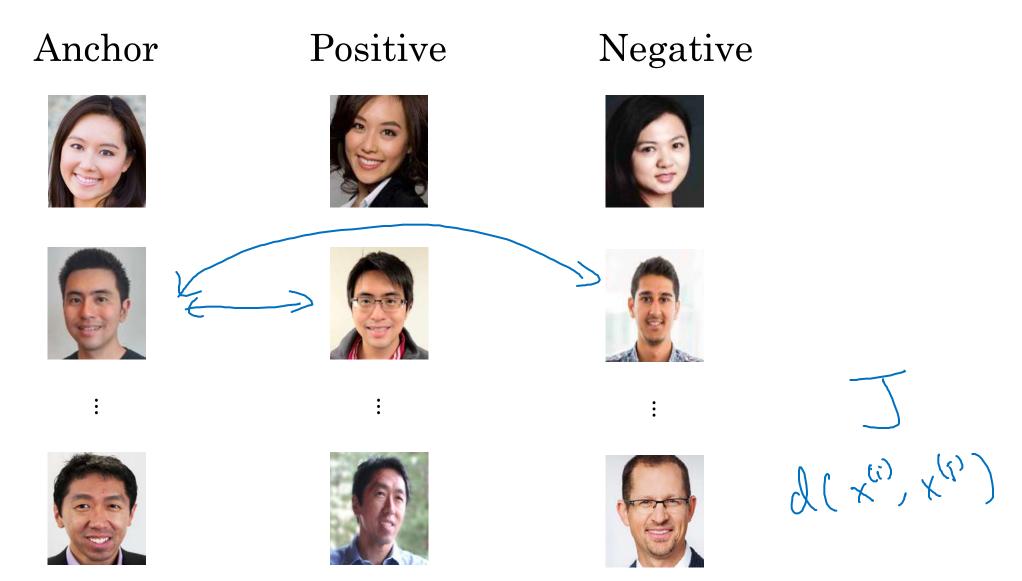
During training, if A,P,N are chosen randomly,  $d(A,P) + \alpha \le d(A,N)$  is easily satisfied.  $\|f(A) - f(P)\|^2 + \alpha \le \|f(A) - f(N)\|^2$ 

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





## Training set using triplet loss

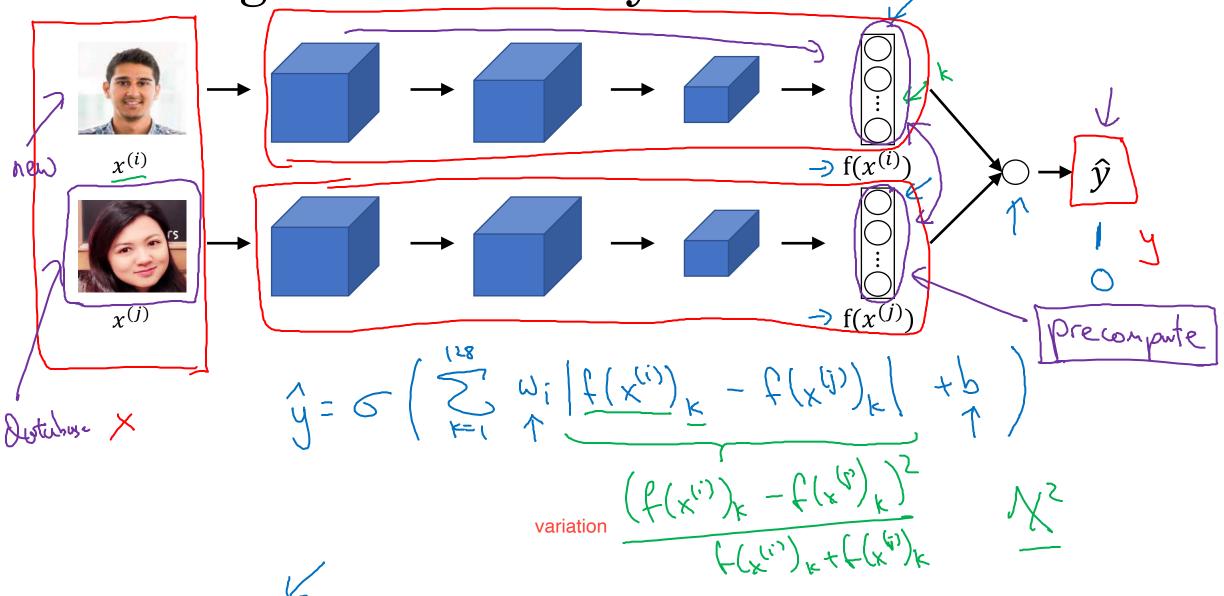




## Face recognition

Face verification and binary classification

Learning the similarity function



## Face verification supervised learning



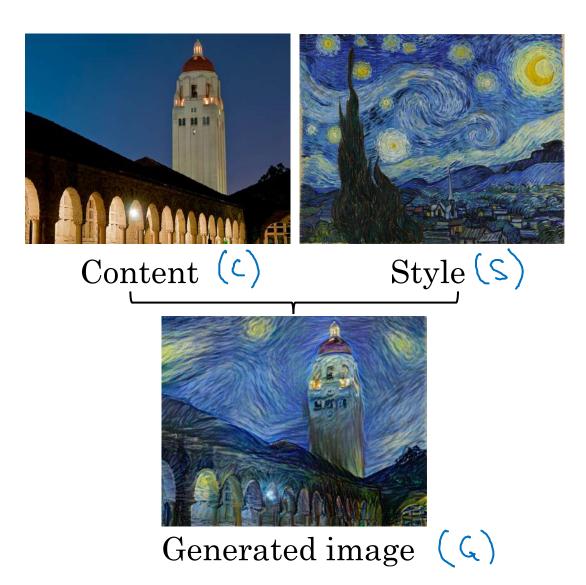
[Taigman et. al., 2014. DeepFace closing the gap to human level performance]



## Neural Style Transfer

What is neural style transfer?

#### Neural style transfer



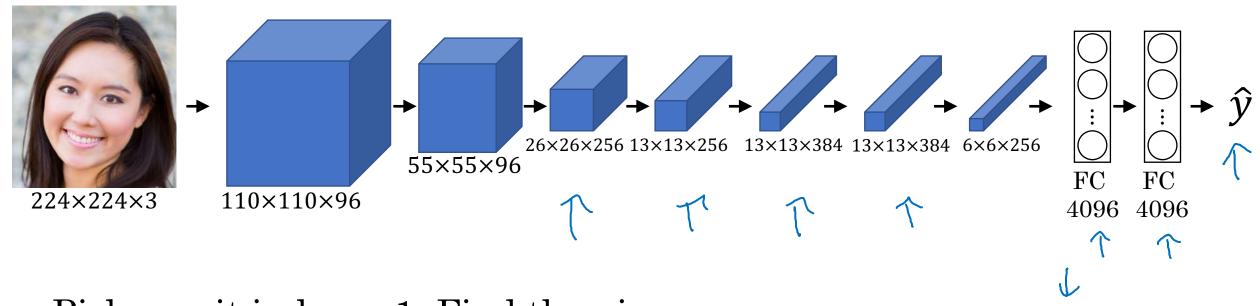
Content () Style Generated image



## Neural Style Transfer

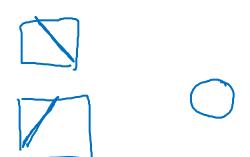
What are deep ConvNets learning?

## 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 2



Layer 3

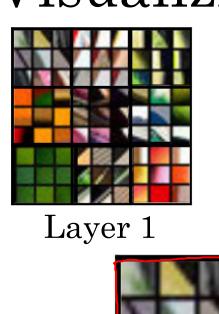


Layer 4



Layer 5

larger image batches in deeper layer









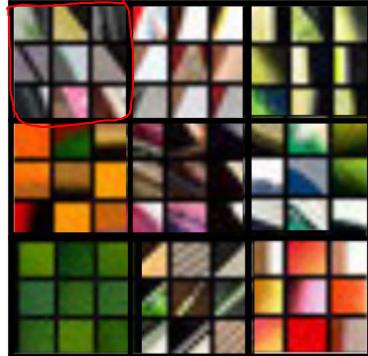


Layer 2

Layer 3

Layer 4

Layer 5











Layer 2



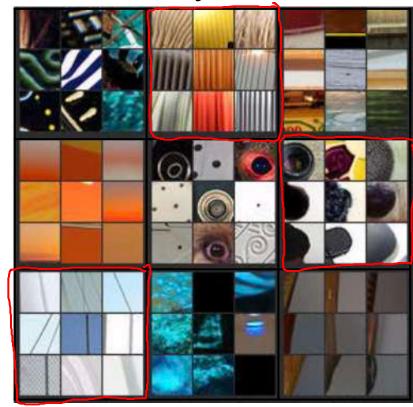
Layer 3



Layer 4



Layer 5





Layer 1



Layer 2



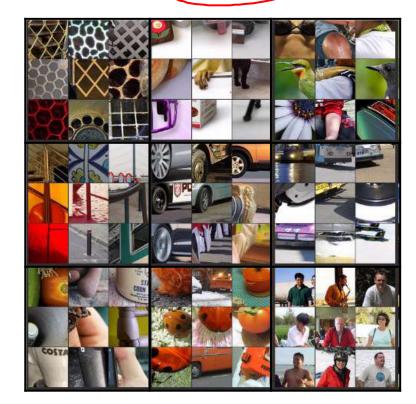
Layer 3



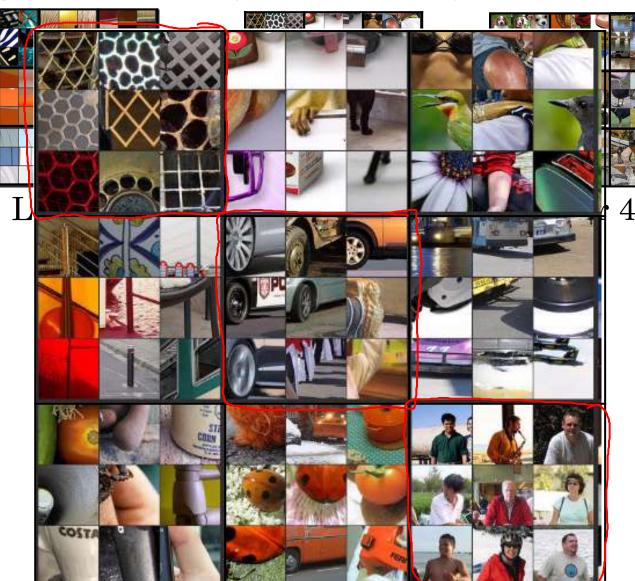
Layer 4



Layer 5

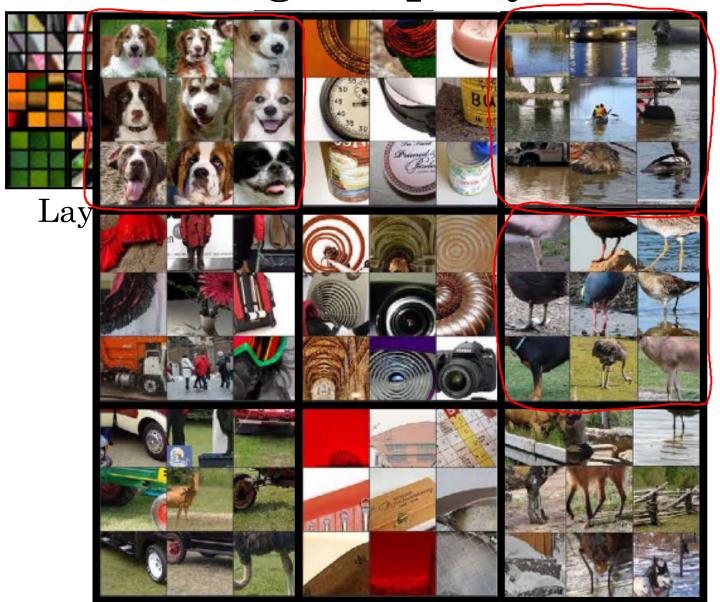








Layer 5





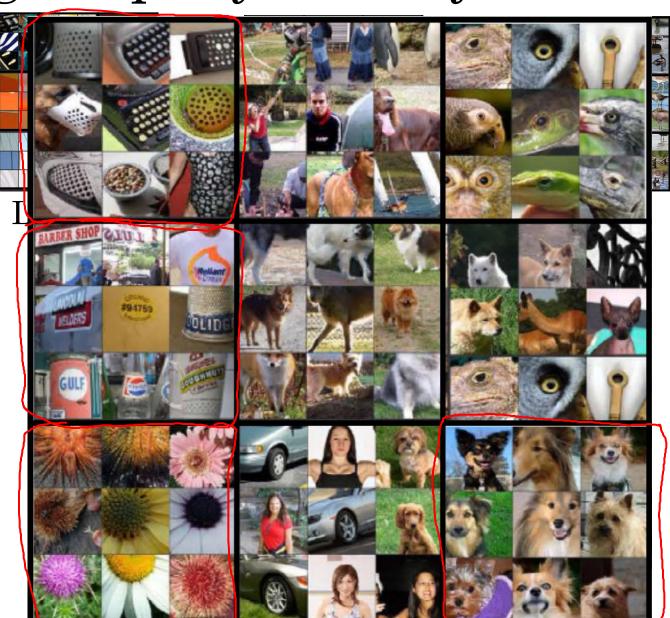
Layer 4



Layer 5









Layer 5



## Neural Style Transfer

## Cost function

#### Neural style transfer cost function



Content C Style S



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

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

content cost (how similar C and G) + style cost (how similar S and G)

## Find the generated image G

1. Initiate G randomly

G: 
$$100 \times 100 \times 3$$

T RUB

2. Use gradient descent to minimize J(G)

$$G:=G-\frac{d}{2G}J(G)$$















## 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  $\underline{a^{[l](C)}}$  and  $\underline{a^{[l](G)}}$  be the activation of layer l on the images
- If  $a^{[l](C)}$  and  $a^{[l](G)}$  are similar, both images have similar content  $\int_{Content} \left( C_{C} C_{C} \right) = \frac{1}{2} \left[ \left( C_{C} C_{C} \right) \left( C_{C} C_{C} C_{C} \right) \right]^{2}$

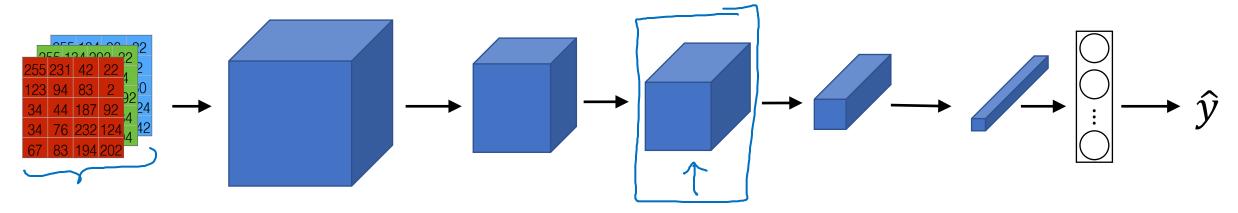
Andrew Ng



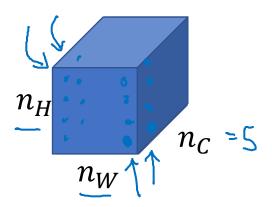
## Neural Style Transfer

## Style cost function

## Meaning of the "style" of an image

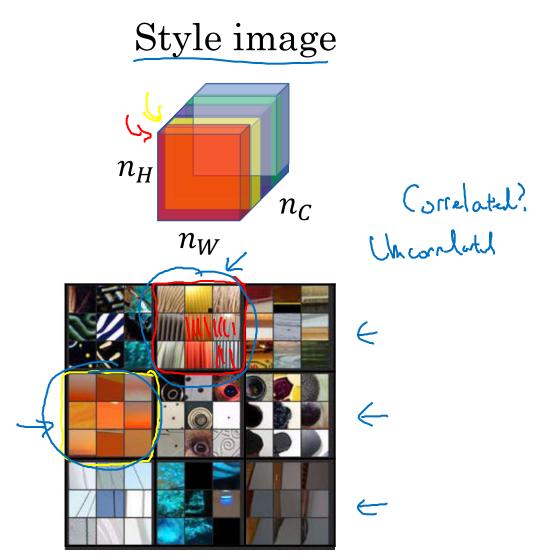


Say you are using layer *l*'s activation to measure "style." Define style as correlation between activations across channels.

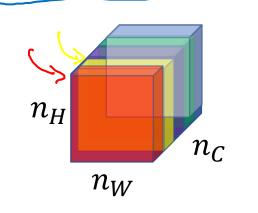


How correlated are the activations across different channels?

#### Intuition about style of an image



#### Generated Image



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

Style matrix

Let 
$$a_{i,j,k}^{[l]} = \text{activation at } (i,j,k)$$
.  $\underline{G}^{[l]} \text{ is } n_{\mathbf{c}}^{[l]} \times n_{\mathbf{c}}^{[l]}$ 

$$\Rightarrow C_{kk'}^{[l]} = \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_{ijk}^{(l)} \alpha_{ijk'}^{(l)}$$

$$\Rightarrow C_{kk'}^{(l)} = \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_{ijk}^{(l)} \alpha_{ijk}^{(l)}$$

$$\int_{S+y}^{CLT} (S, G) = \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G) - G_{2}(G) \right\|_{F}^{2}$$

$$= \frac{1}{(S-1)} \left\| G_{1}(S) - G_{2}(G) - G_{2}(G)$$

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

#### Style cost function

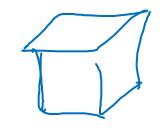
$$J_{style}^{[l]}(S,G) = \frac{1}{\left(2n_H^{[l]}n_W^{[l]}n_C^{[l]}\right)^2} \sum_{k} \sum_{k'} (G_{kk'}^{[l](S)} - G_{kk'}^{[l](G)})$$

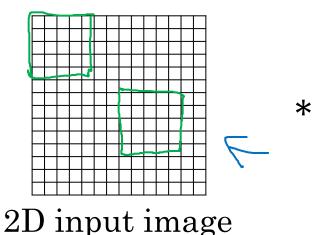


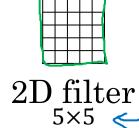
# Convolutional Networks in 1D or 3D

1D and 3D generalizations of models

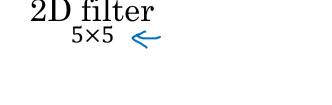
#### Convolutions in 2D and 1D

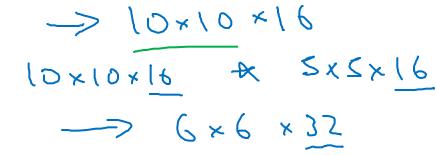


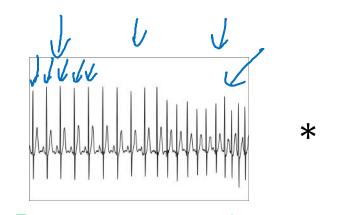












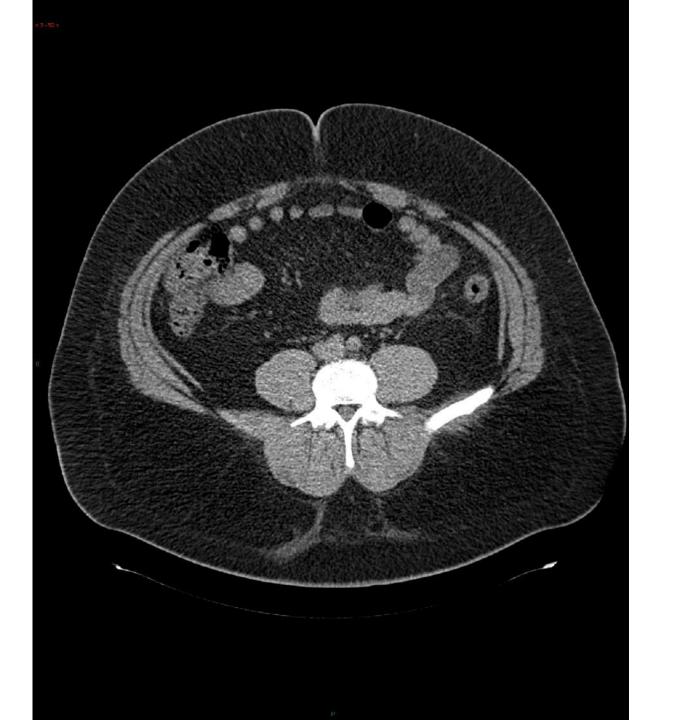
14×14 <--



10

14	× \	*	5 × 1
		•	















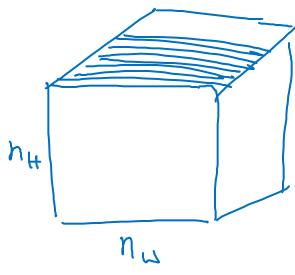












#### 3D convolution

