



deeplearning.ai

# Neural Style Transfer

---

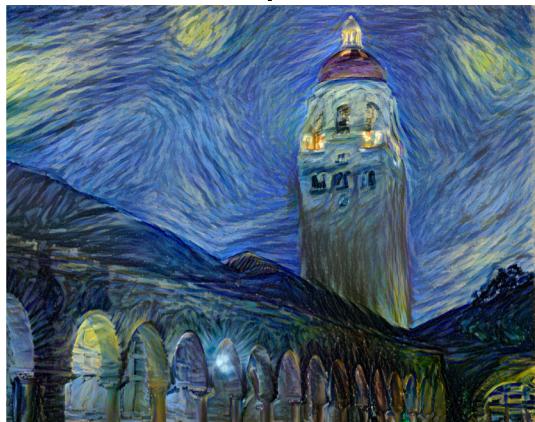
What is neural style  
transfer?

# Neural style transfer



Content ( $c$ )

Style ( $s$ )



Generated image ( $G$ )



Content ( $c$ )

Style ( $s$ )



Generated image ( $G$ )



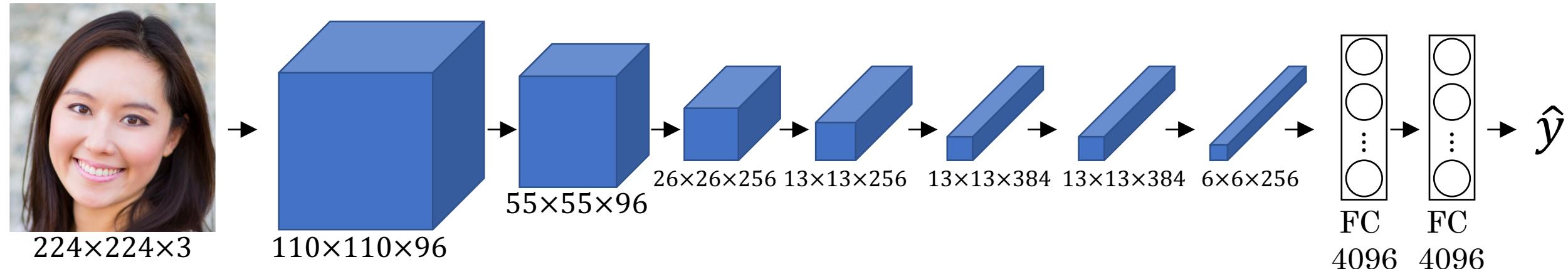
deeplearning.ai

# Neural Style Transfer

---

What are deep  
ConvNets learning?

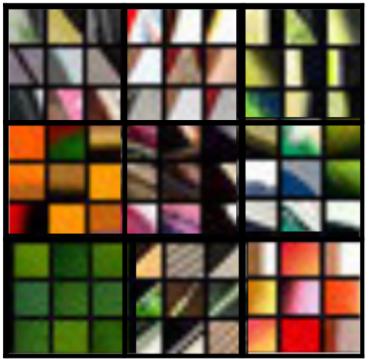
# Visualizing what a deep network is learning



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

Repeat for other units.

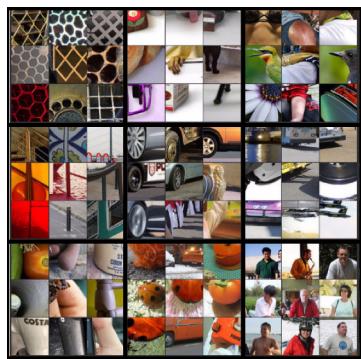
# Visualizing deep layers



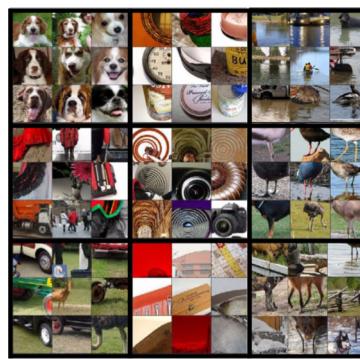
Layer 1



Layer 2



Layer 3

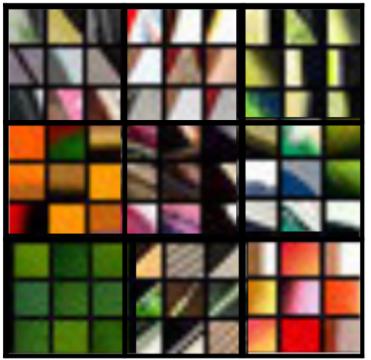


Layer 4

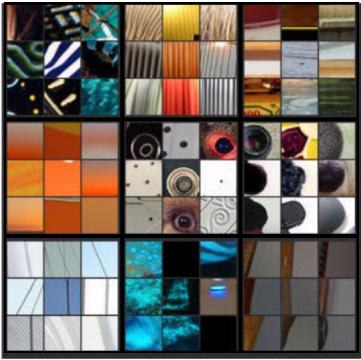


Layer 5

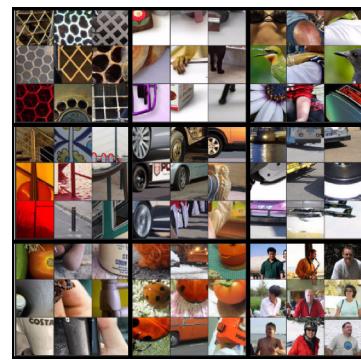
# Visualizing deep layers: Layer 1



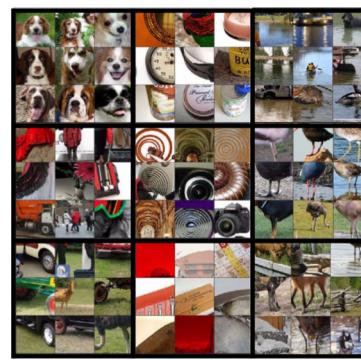
Layer 1



Layer 2



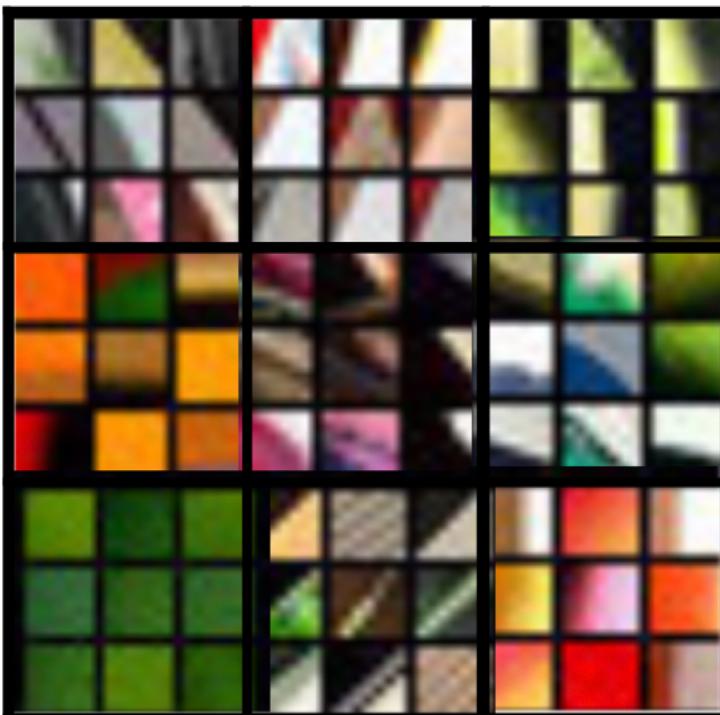
Layer 3



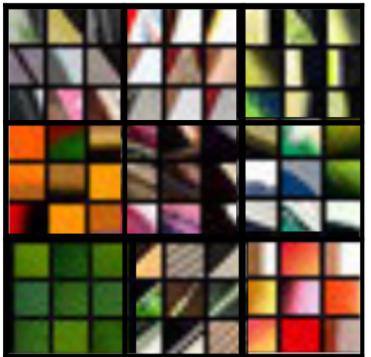
Layer 4



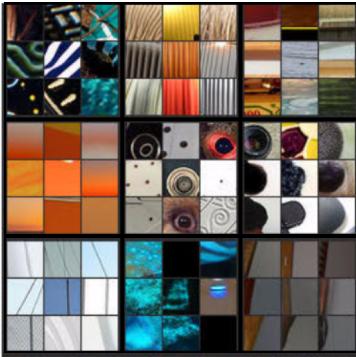
Layer 5



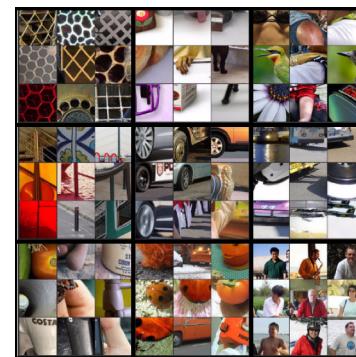
# Visualizing deep layers: Layer 2



Layer 1



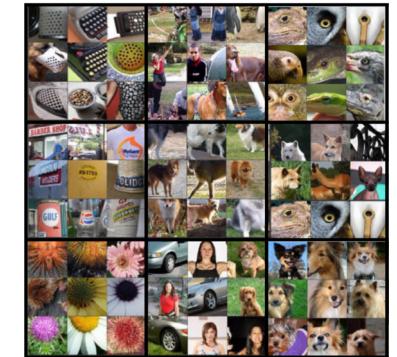
Layer 2



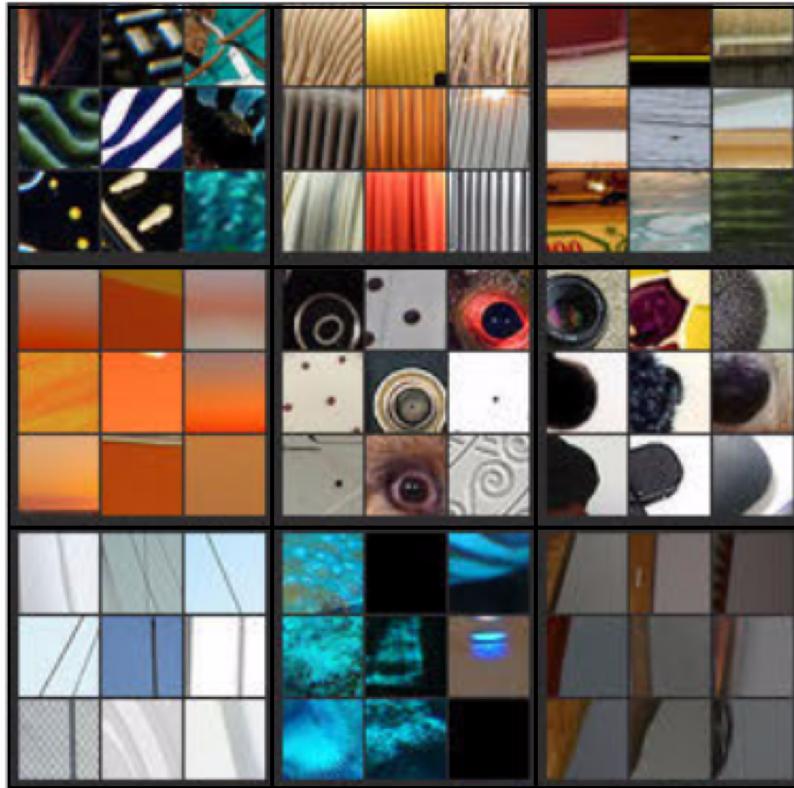
Layer 3



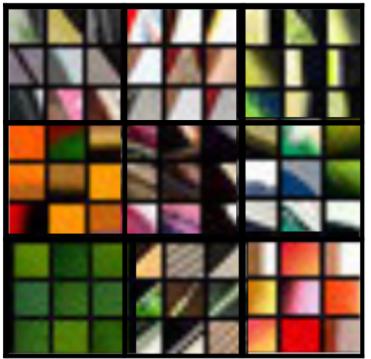
Layer 4



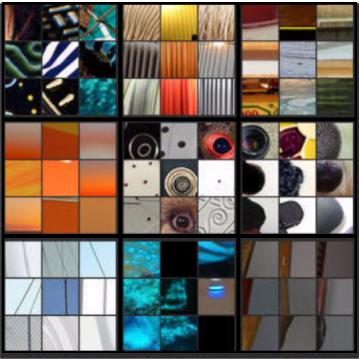
Layer 5



# Visualizing deep layers: Layer 3



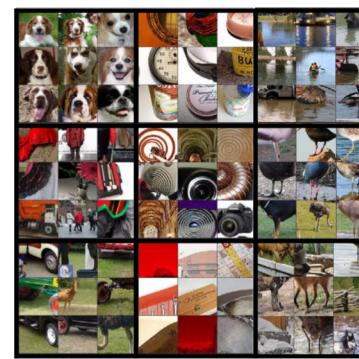
Layer 1



Layer 2



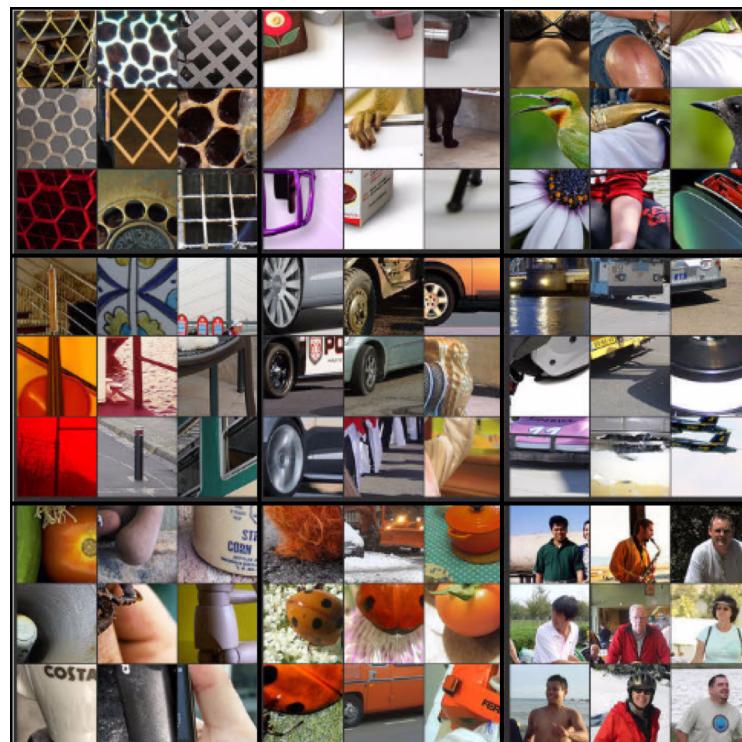
Layer 3



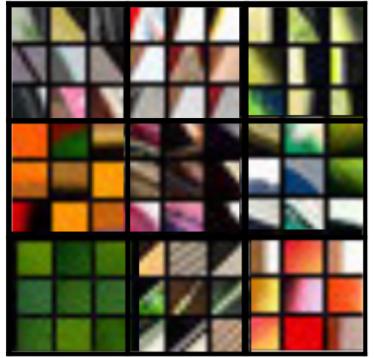
Layer 4



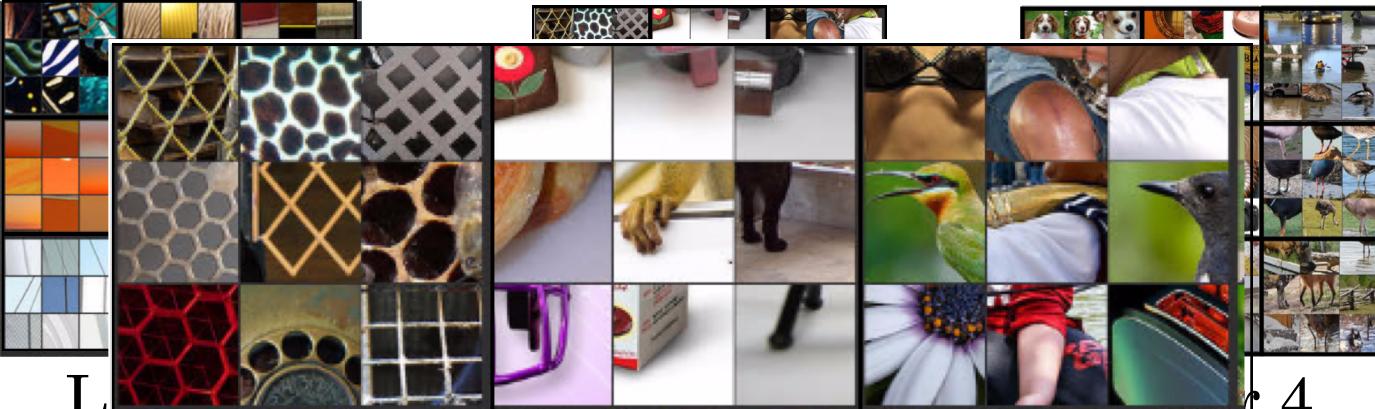
Layer 5



# Visualizing deep layers: Layer 3



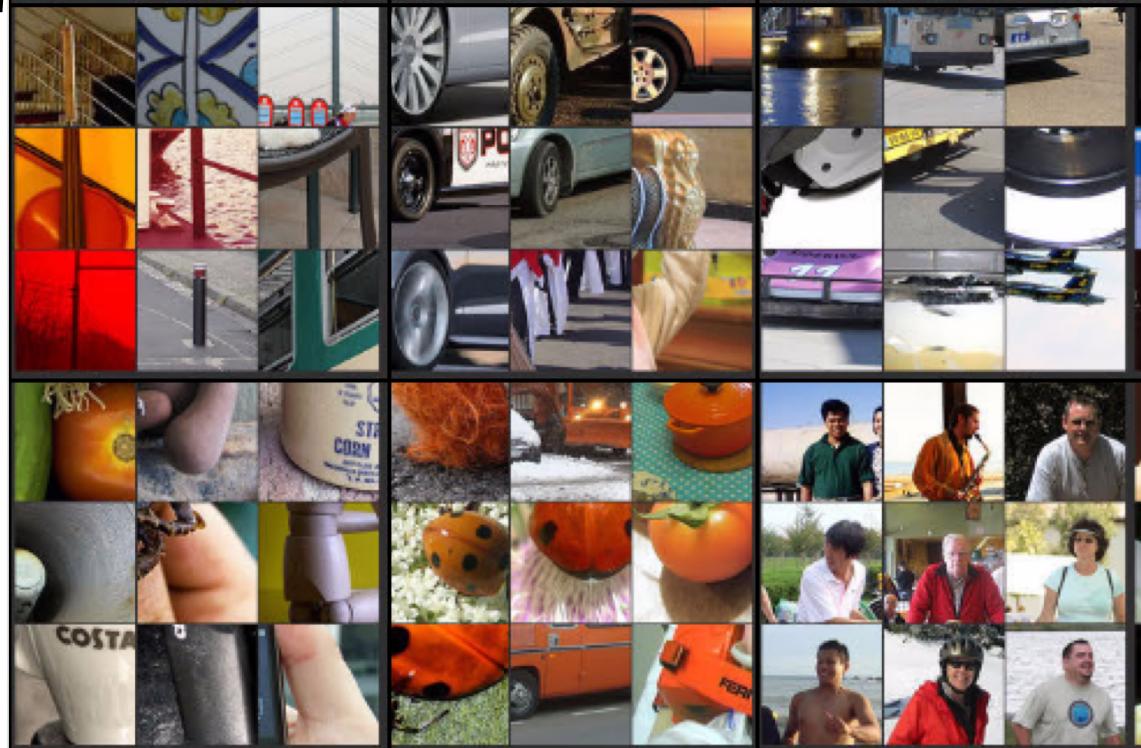
Layer 1



L

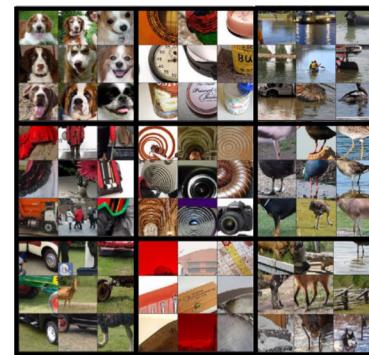
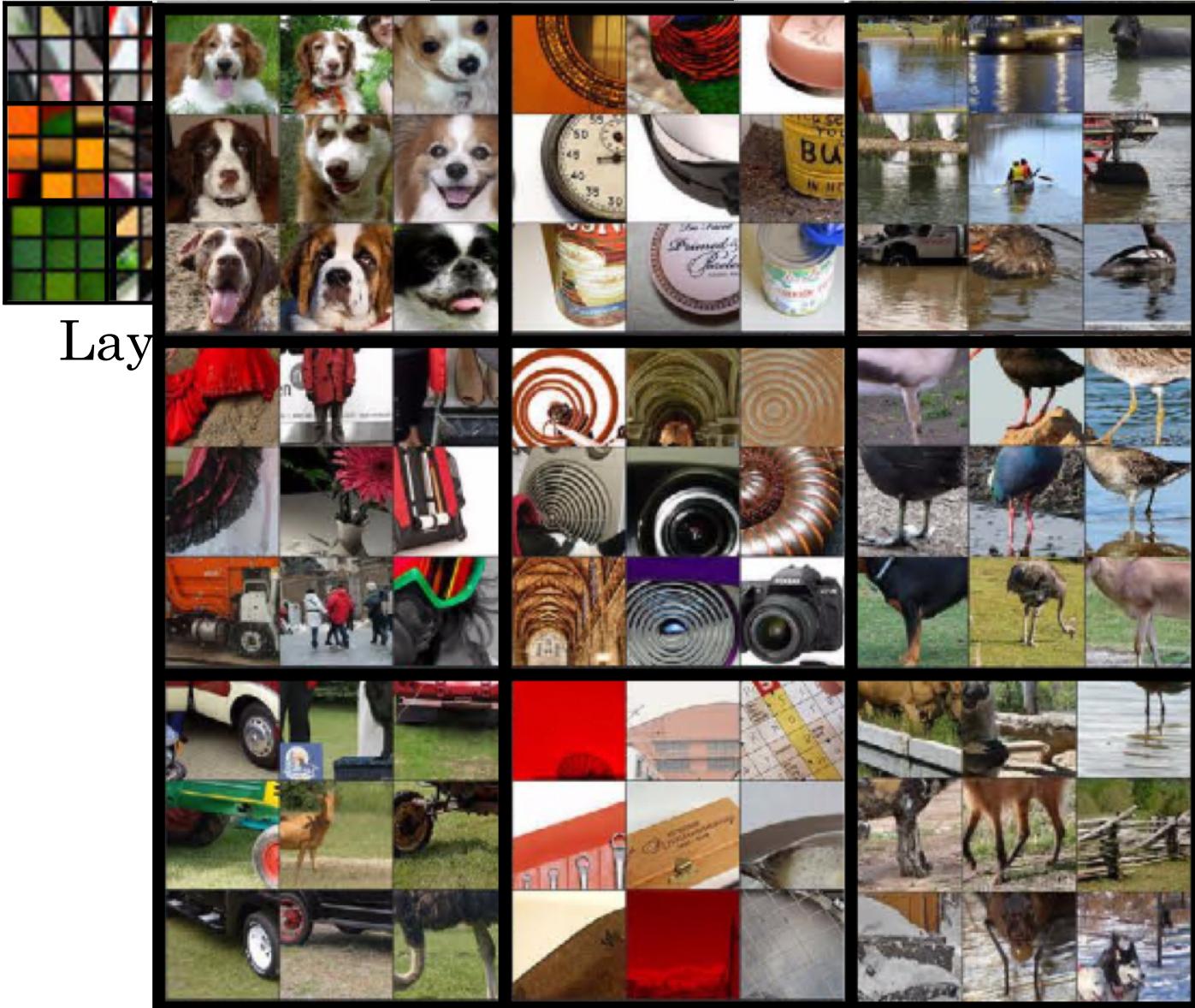


Layer 5

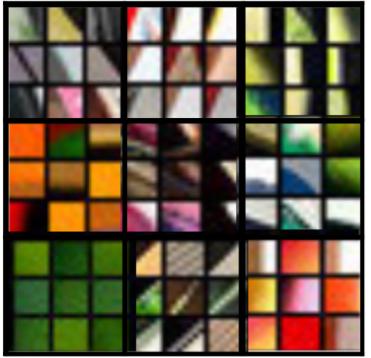


• 4

# Visualizing deep layers: Layer 4



# Visualizing deep layers: Layer 5



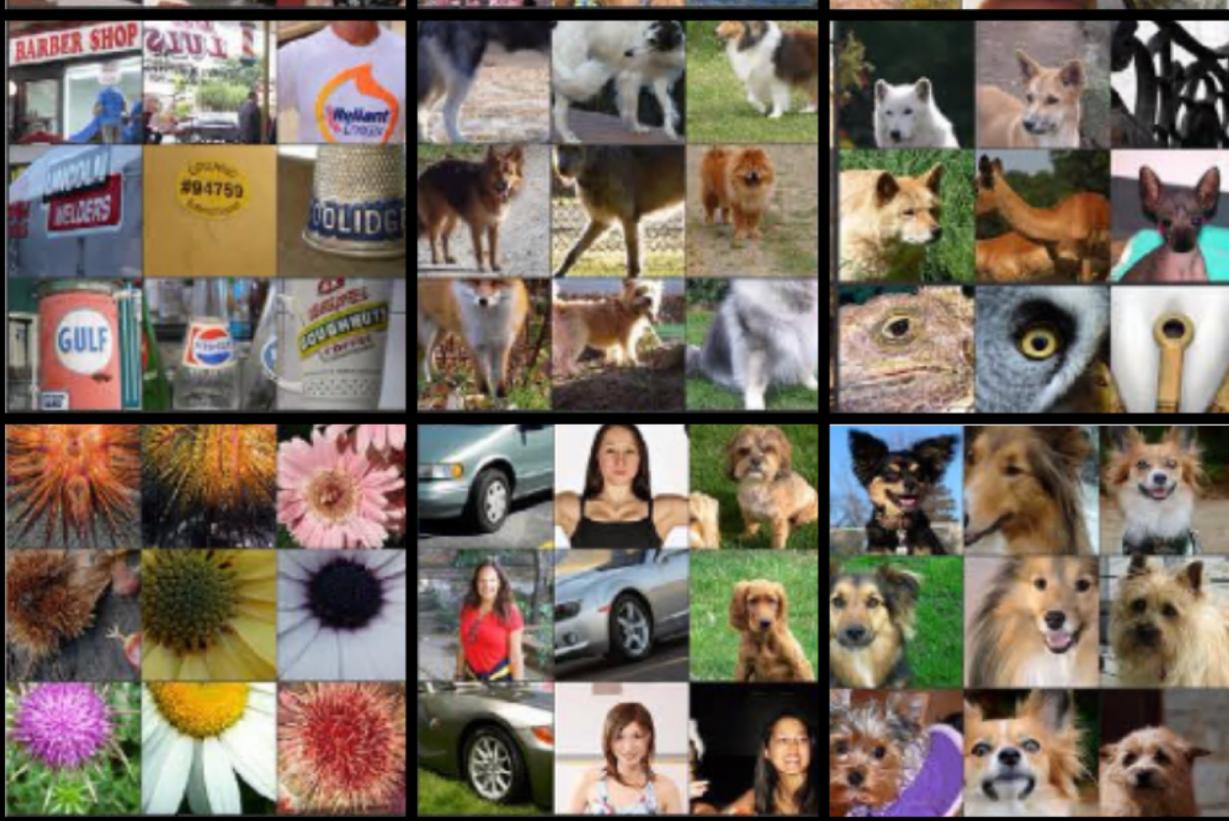
Layer 1



Layer 2



Layer 3



Layer 4



Layer 5



deeplearning.ai

# Neural Style Transfer

---

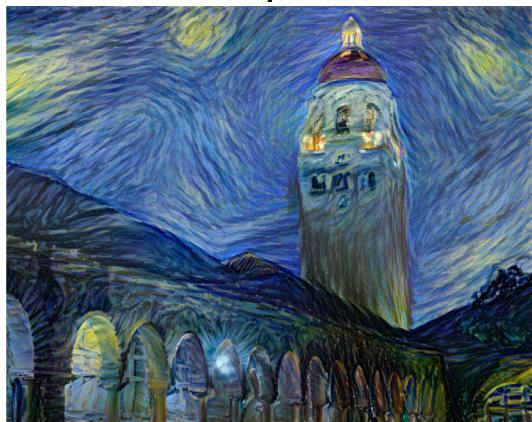
## Cost function

# Neural style transfer cost function



Content C

Style S



Generated image G

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

# Find the generated image $G$

1. Initiate  $G$  randomly

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

2. Use gradient descent to minimize  $\underline{J(G)}$

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





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  $\underline{a}^{[l](C)}$  and  $\underline{a}^{[l](G)}$  are similar, both images have similar content

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



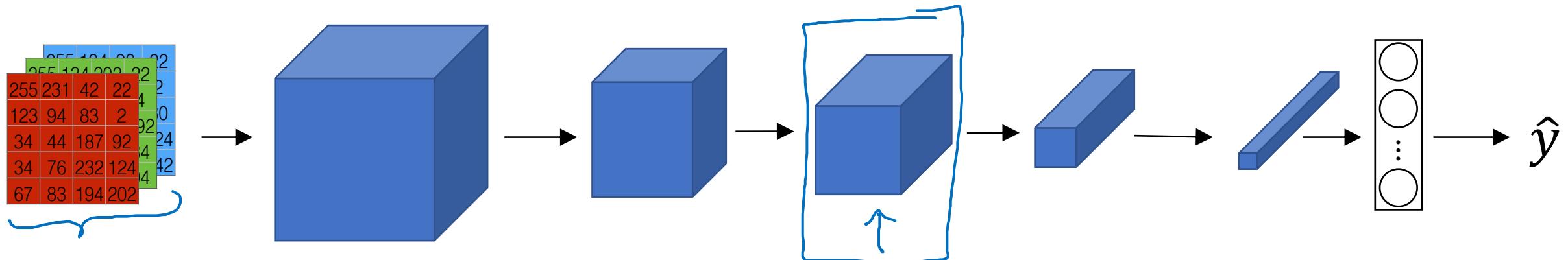
deeplearning.ai

# Neural Style Transfer

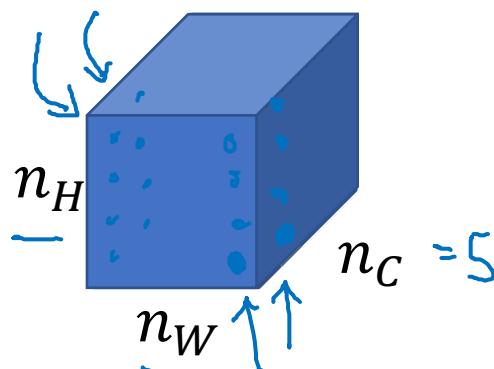
---

## Style cost function

# Meaning of the “style” of an image

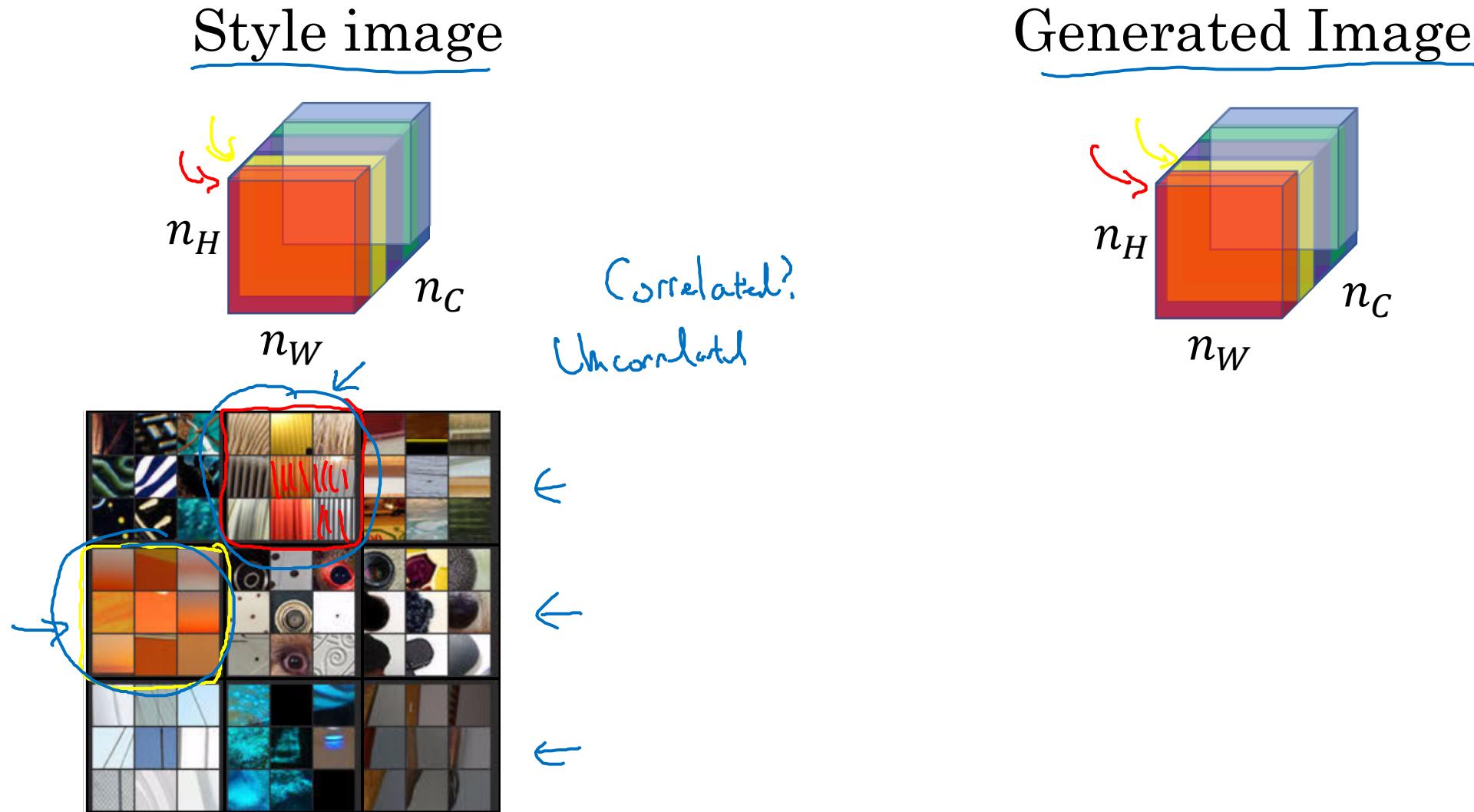


Say you are using layer  $l$ 's activation to measure “style.”  
Define style as correlation between activations across channels.



How correlated are the activations  
across different channels?

# Intuition about style of an image



# Style matrix

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

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

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

H W C  
↓ ↓ ↓

$n_c$

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

"Gram matrix"

$$\begin{aligned} J_{style}^{[l]}(S, G) &= \frac{1}{(\dots)} \| G_{kk'}^{[l](s)} - G_{kk'}^{[l](w)} \|_F^2 \\ &= \frac{1}{(2n_h^{[l]} n_w^{[l]})^2} \sum_k \sum_{k'} (G_{kk'}^{[l](s)} - G_{kk'}^{[l](w)})^2 \end{aligned}$$

# Style cost function

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

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

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

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

G