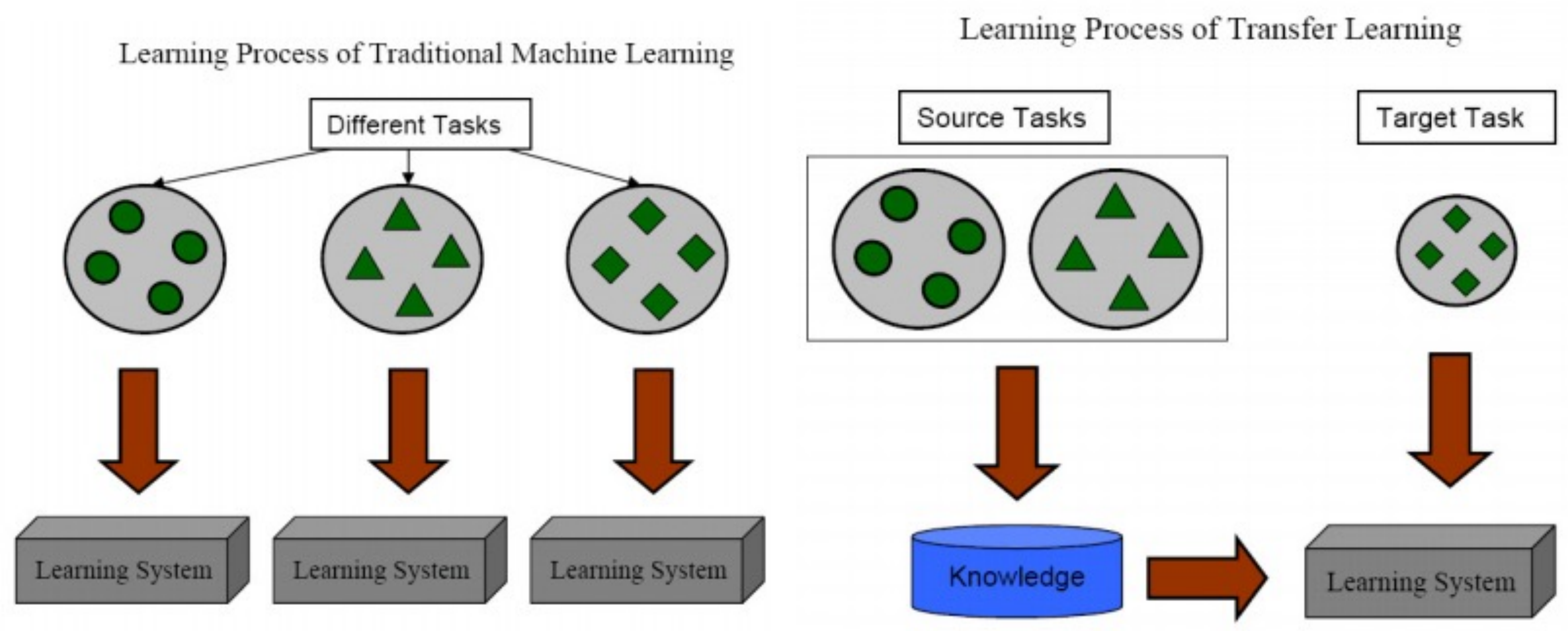
Transfer Learning

Aprendizaje Automático Aplicado

Julio Waissman

Encuentro Estatal de Innovación Tecnológica 2024

Idea básica de transfer learning



(a) Traditional Machine Learning (b) Transfer Learning

Tipos de transfer learning

Inductivo

- Adaptar modelo existente a nuevos datos etiquetados

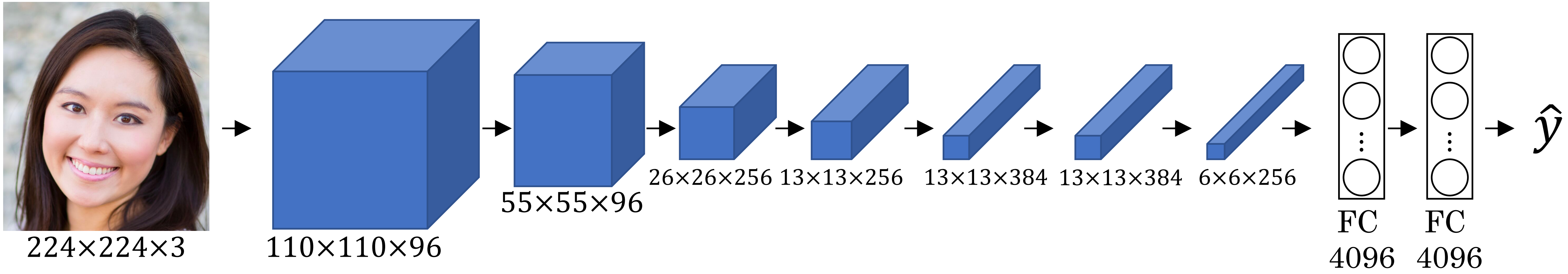
Transductivo

- Adaptar modelo existente a nuevos datos no etiquetados

No supervisado

- Adaptar modelo no supervisado existente a nuevos datos no etiquetados

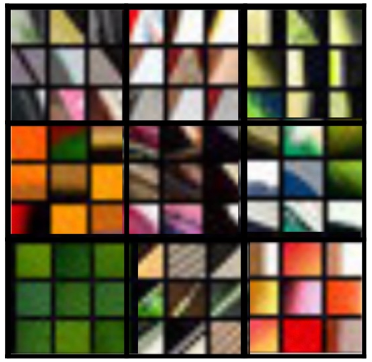
¿Qué aprende una CNN?



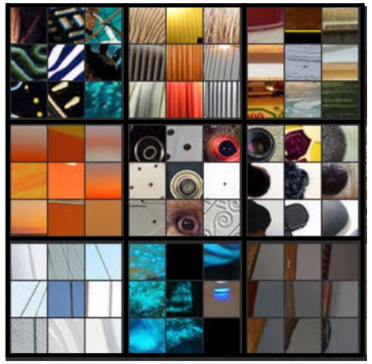
- Tomamos 9 unidades por capa al azar
- Buscamos los 9 parches de muchas imágenes (en la imagen original) que maximice la activación

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

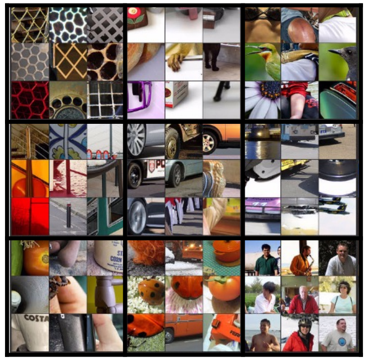
Visualizing deep layers



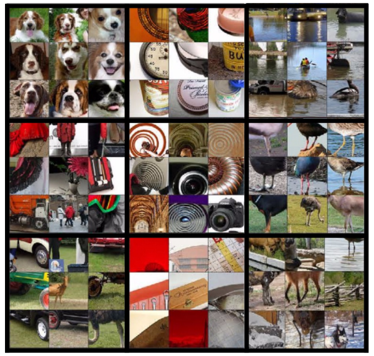
Layer 1



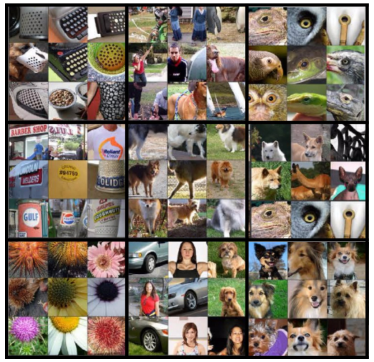
Layer 2



Layer 3

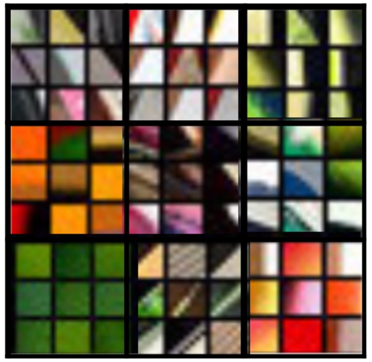


Layer 4

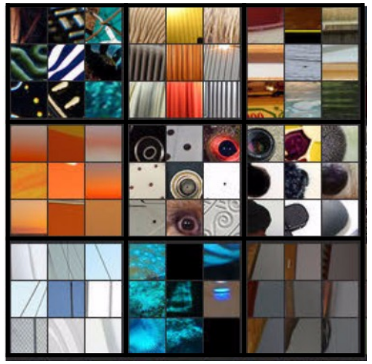


Layer 5

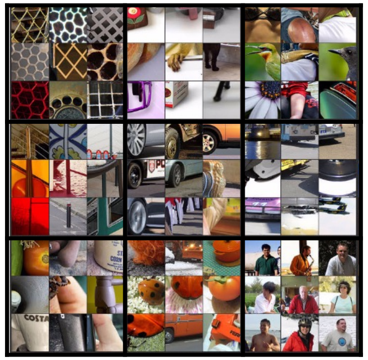
Visualizing deep layers: Layer 1



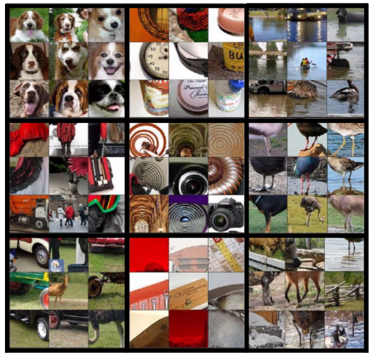
Layer 1



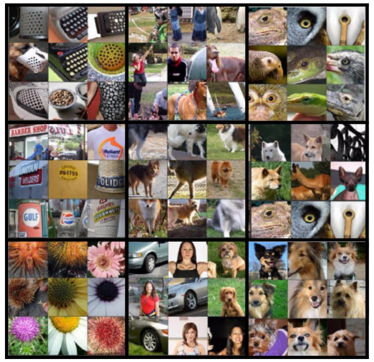
Layer 2



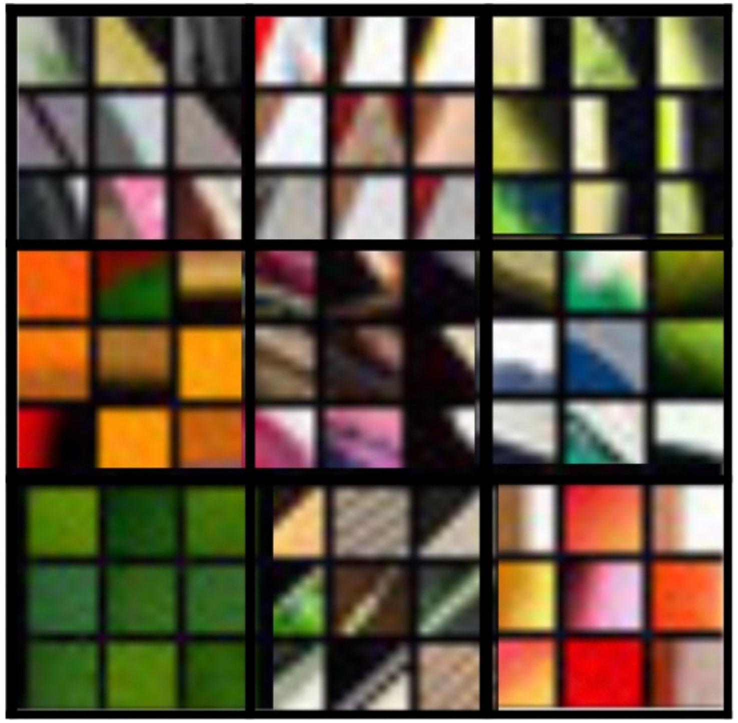
Layer 3



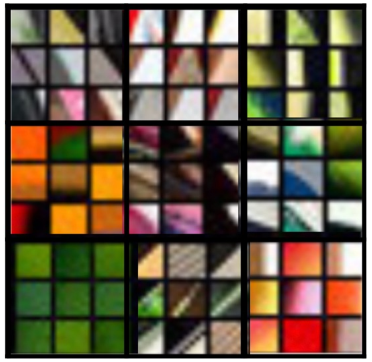
Layer 4



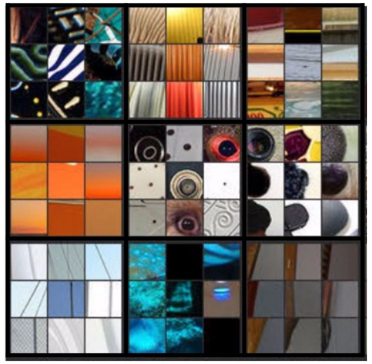
Layer 5



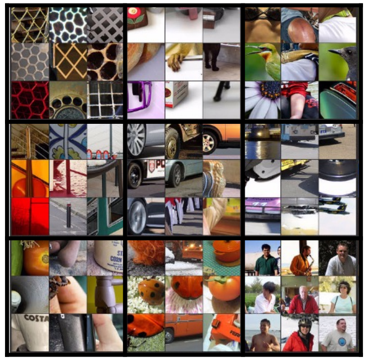
Visualizing deep layers: Layer 2



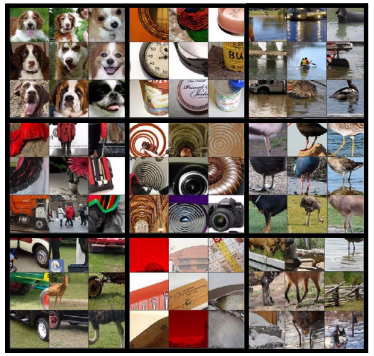
Layer 1



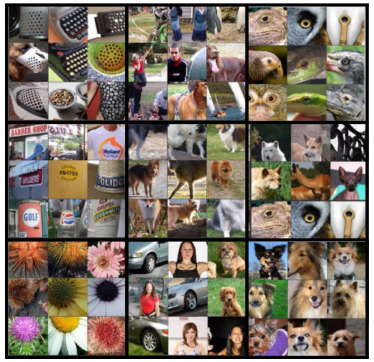
Layer 2



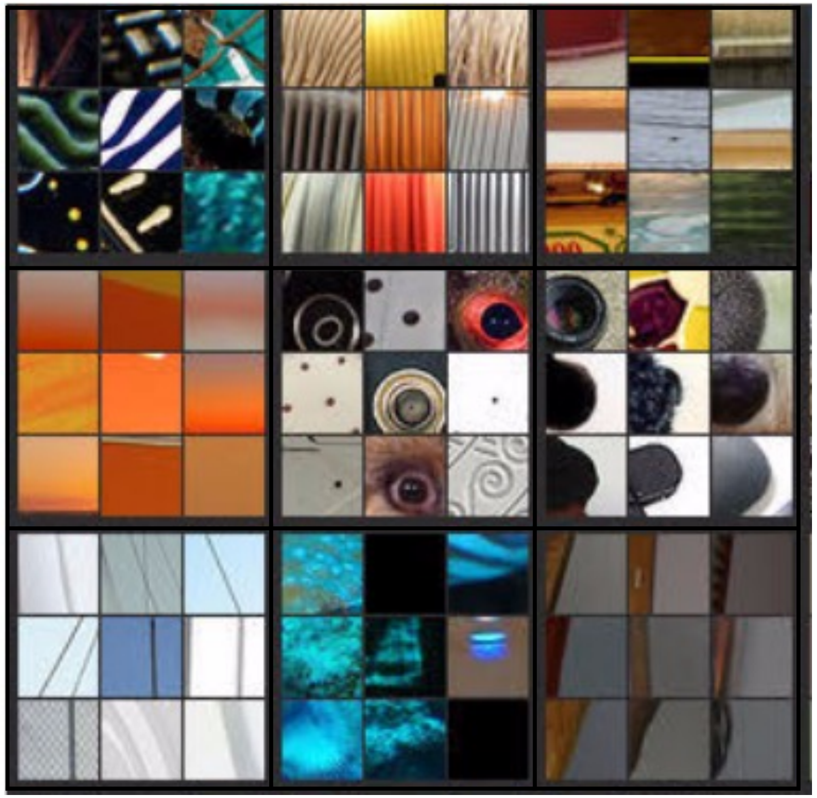
Layer 3



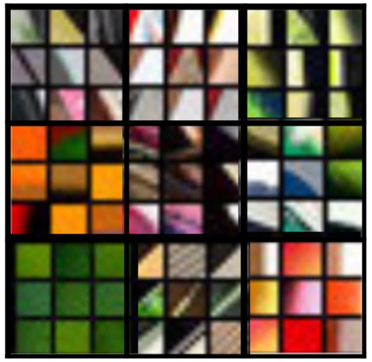
Layer 4



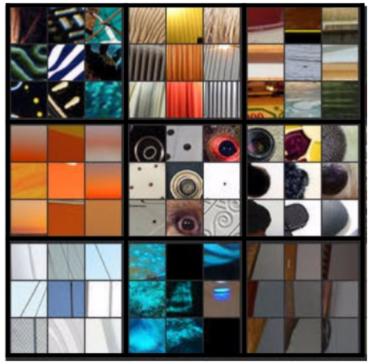
Layer 5



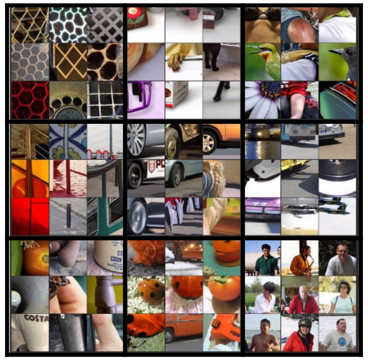
Visualizing deep layers: Layer 3



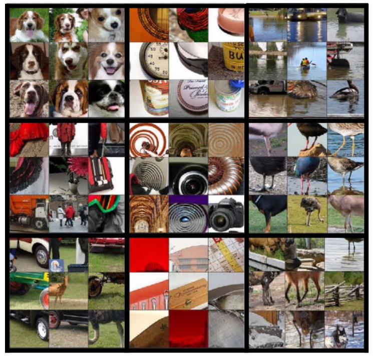
Layer 1



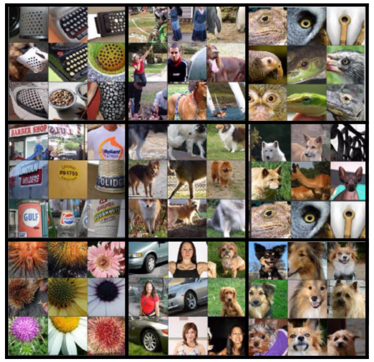
Layer 2



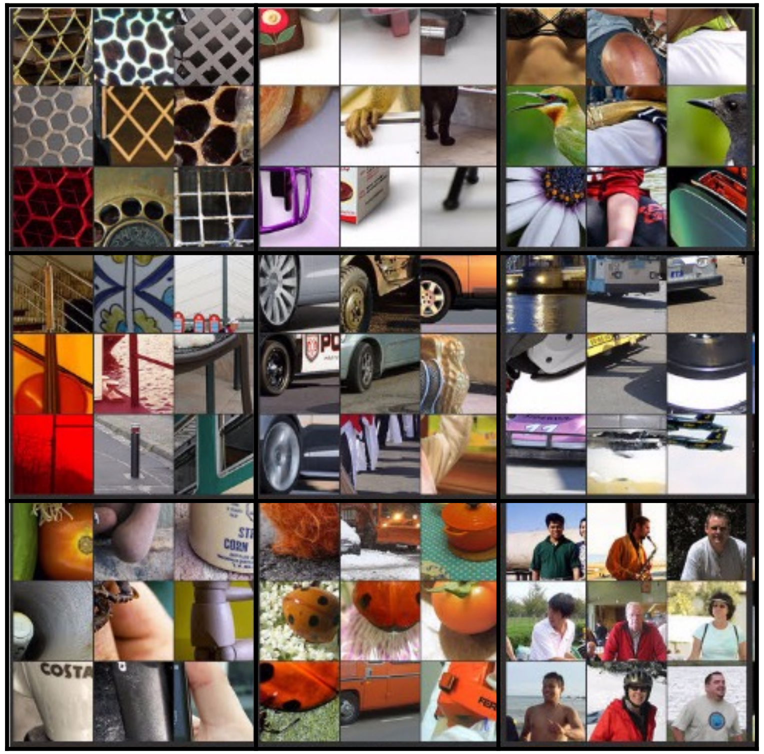
Layer 3



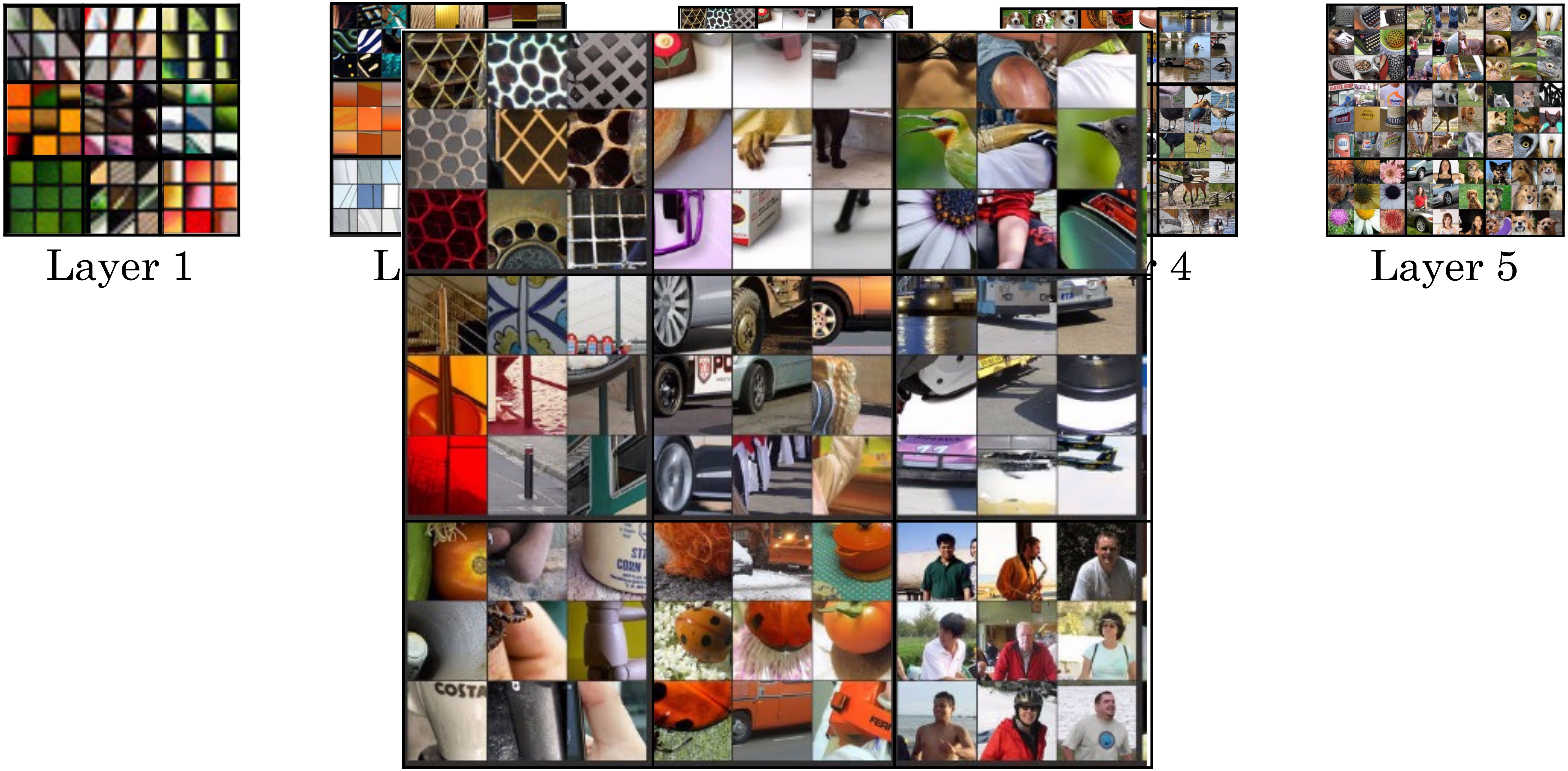
Layer 4



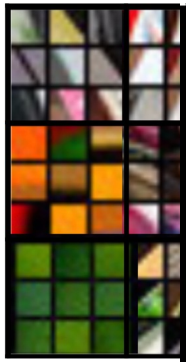
Layer 5



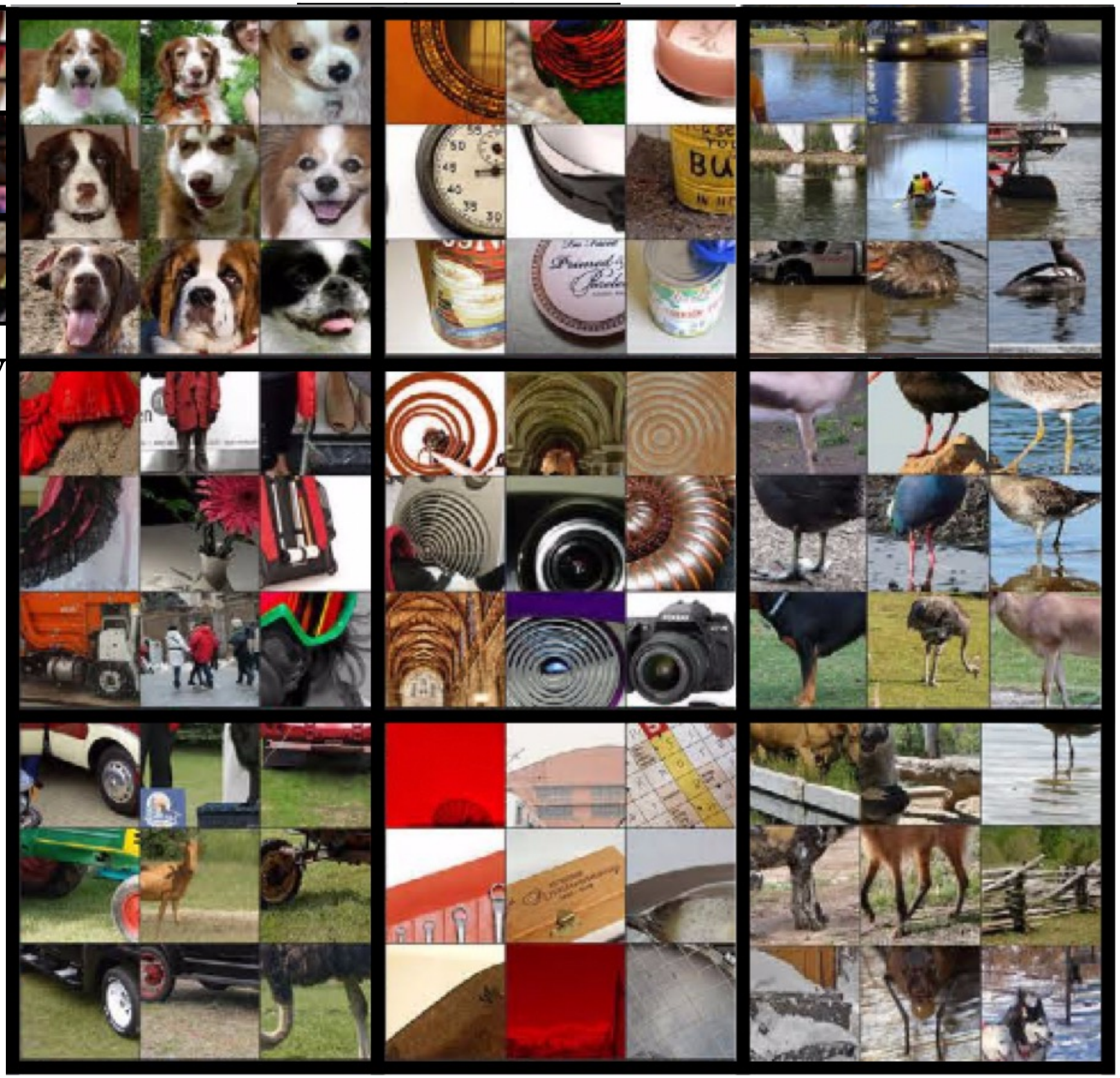
Visualizing deep layers: Layer 3

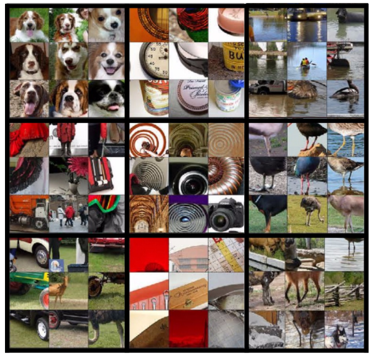


Visualizing deep layers: Layer 4

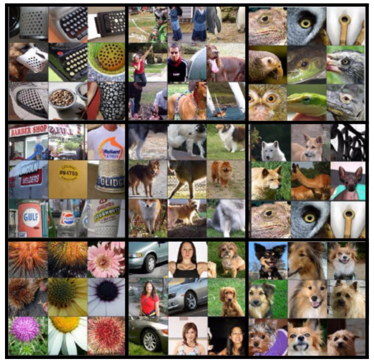


Lay



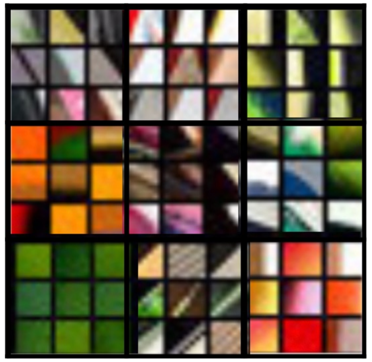


Layer 4

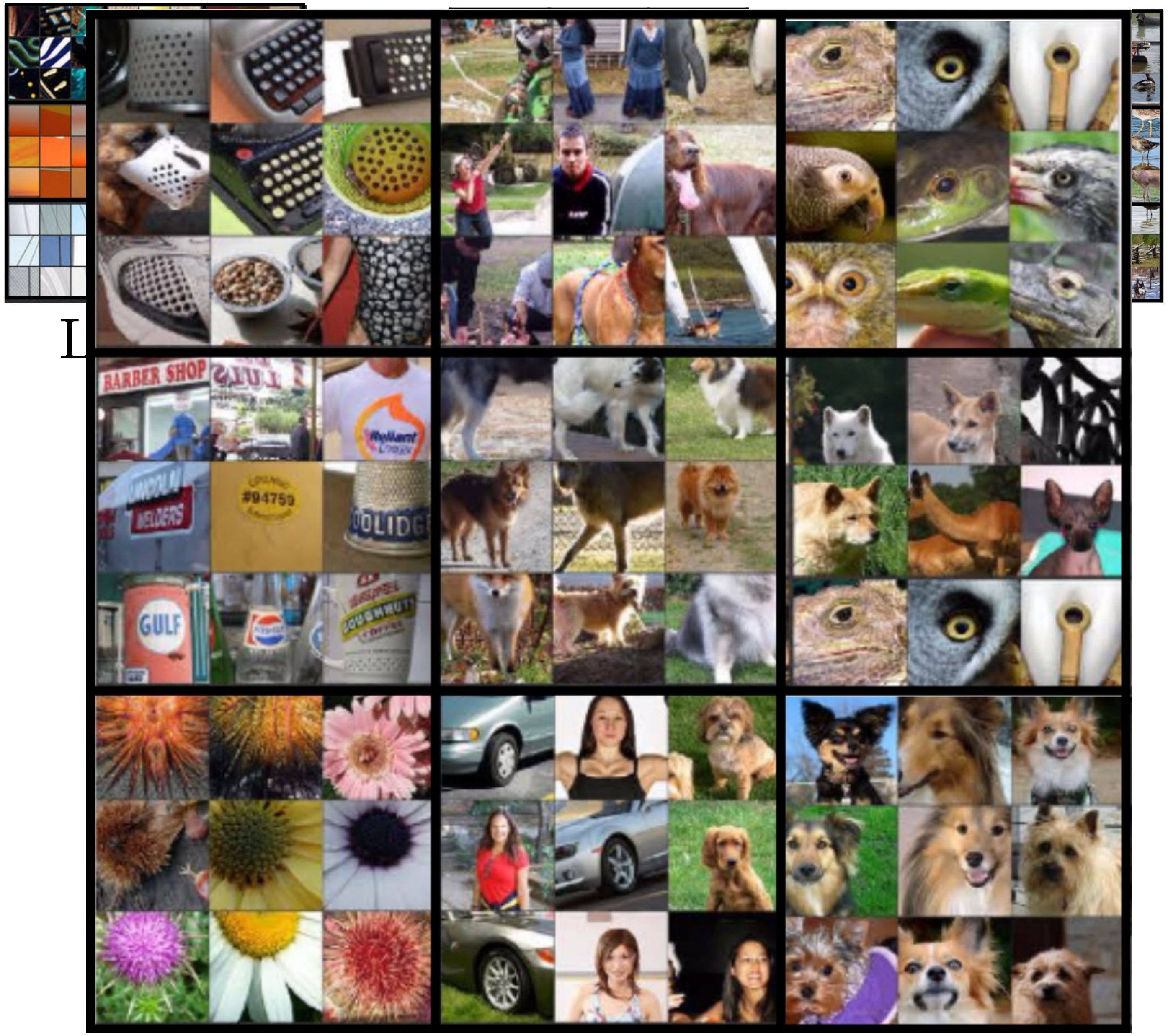


Layer 5

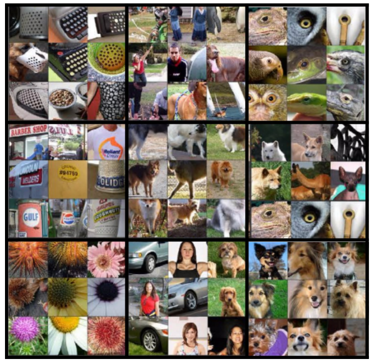
Visualizing deep layers: Layer 5



Layer 1

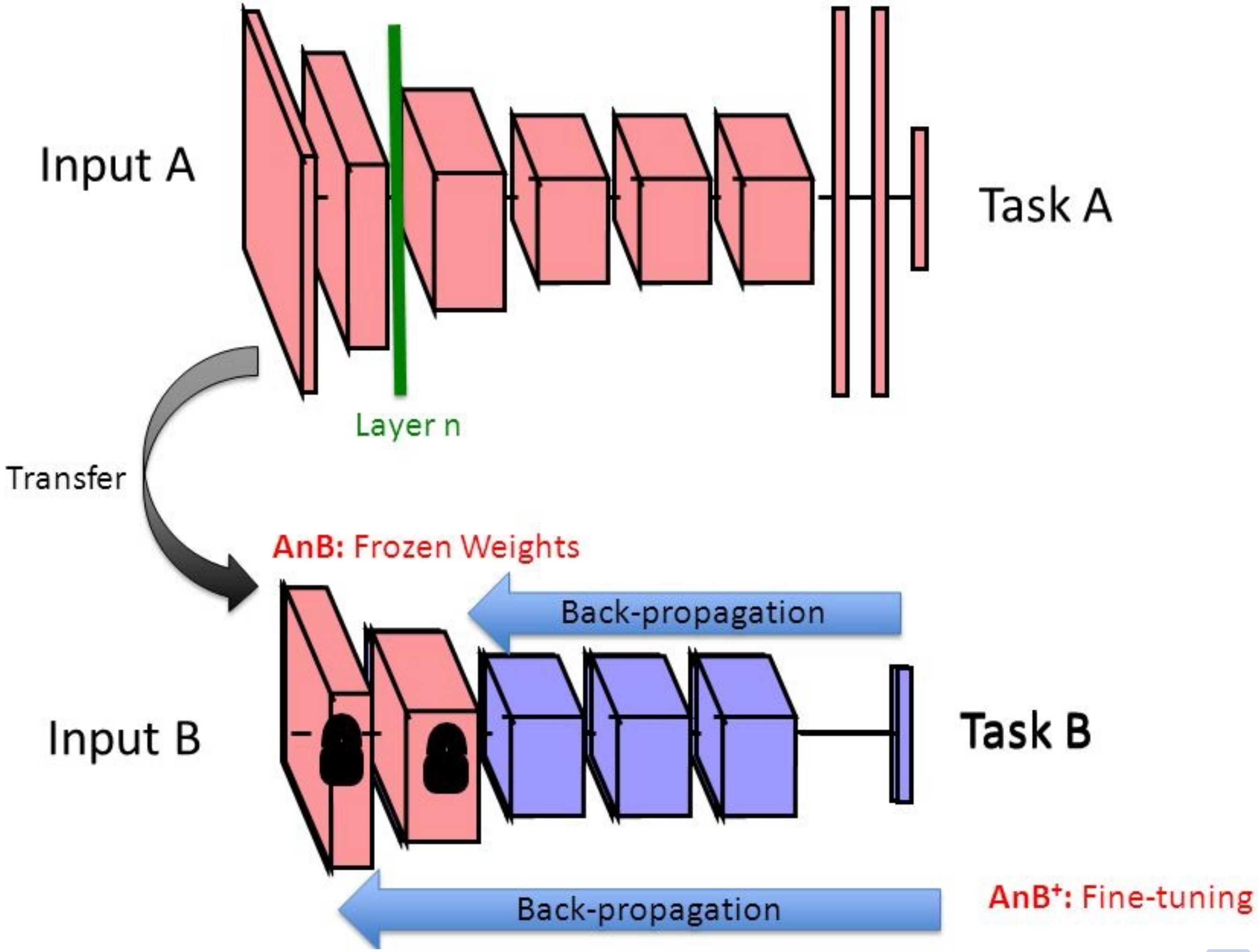


L

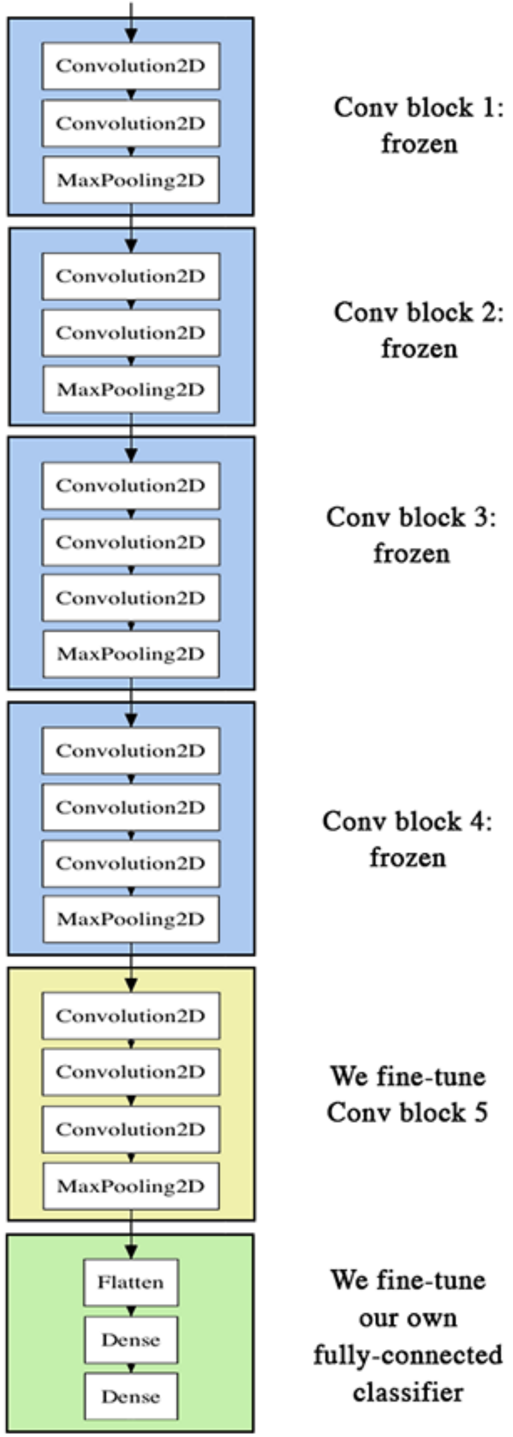


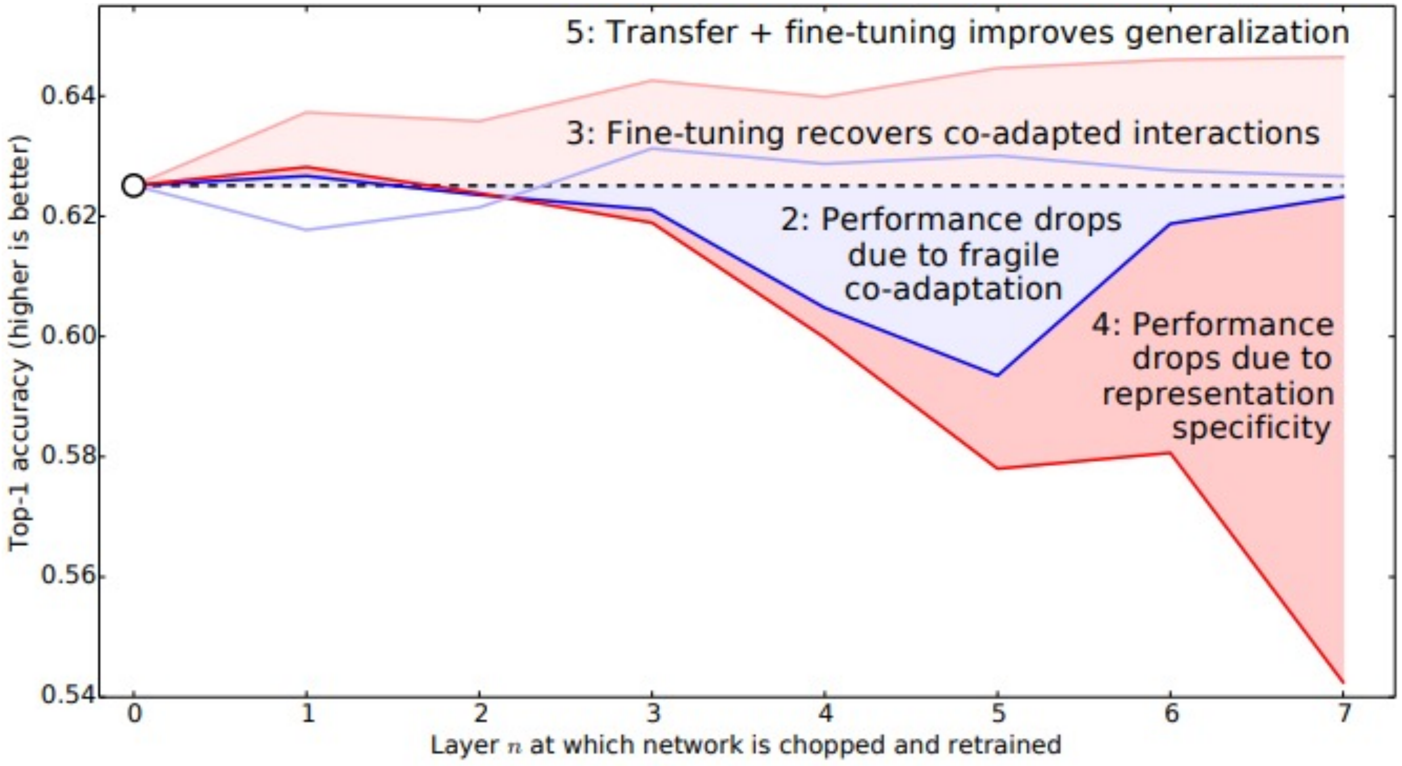
Layer 5

Transfer Learning Overview

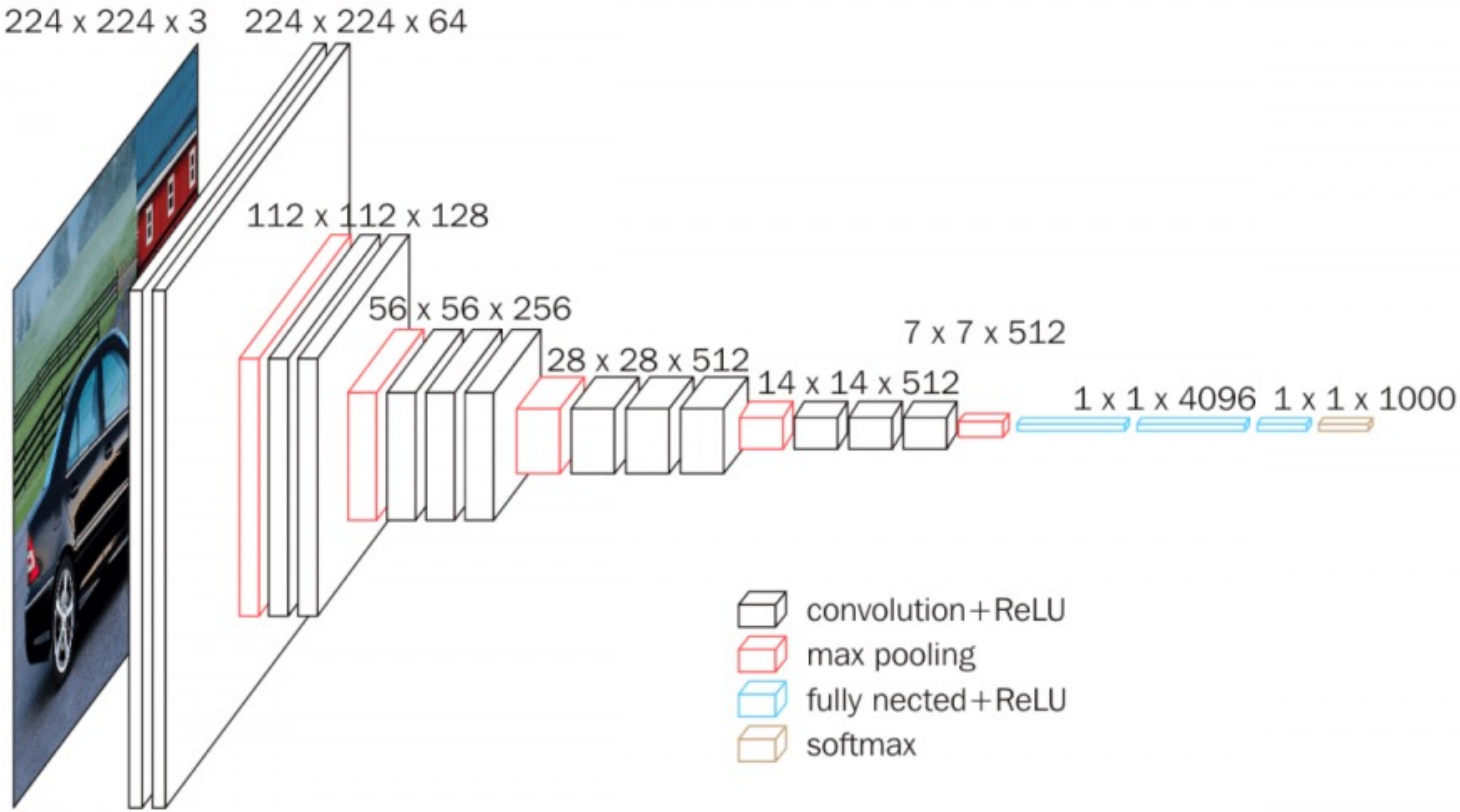


¿Cómo hacer el transfer learning?

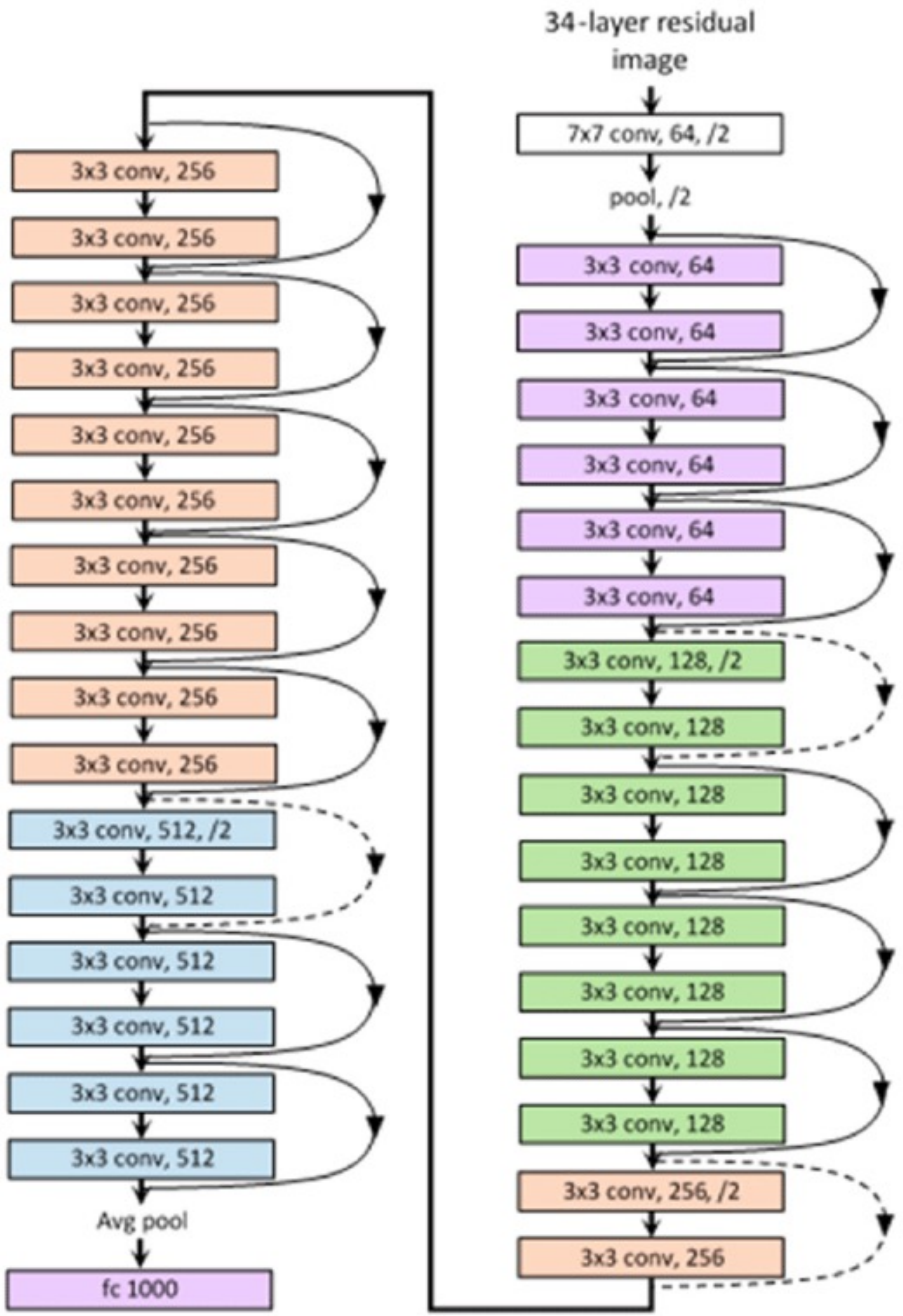




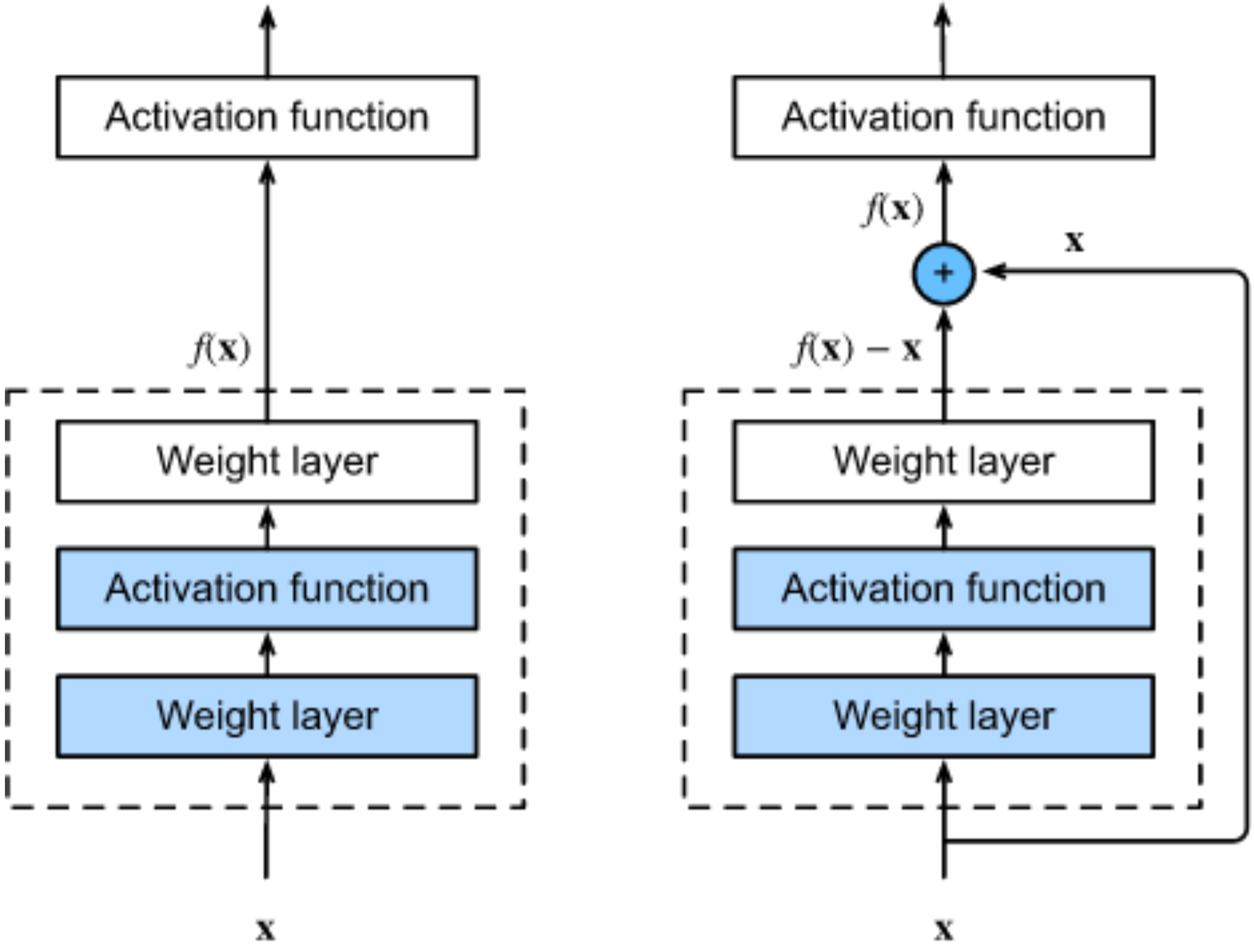
Algunos modelos conocidos: VGG-16

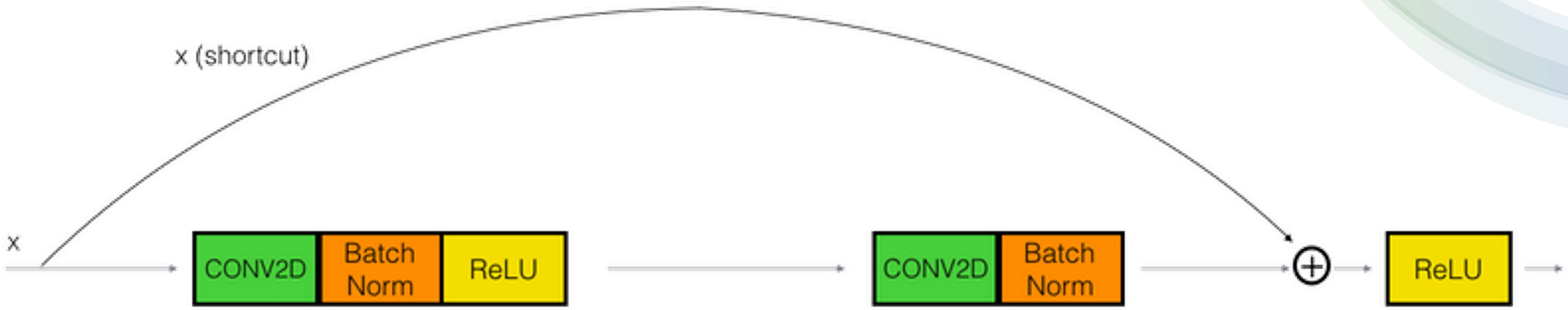


Algunos modelos conocidos: ResNet



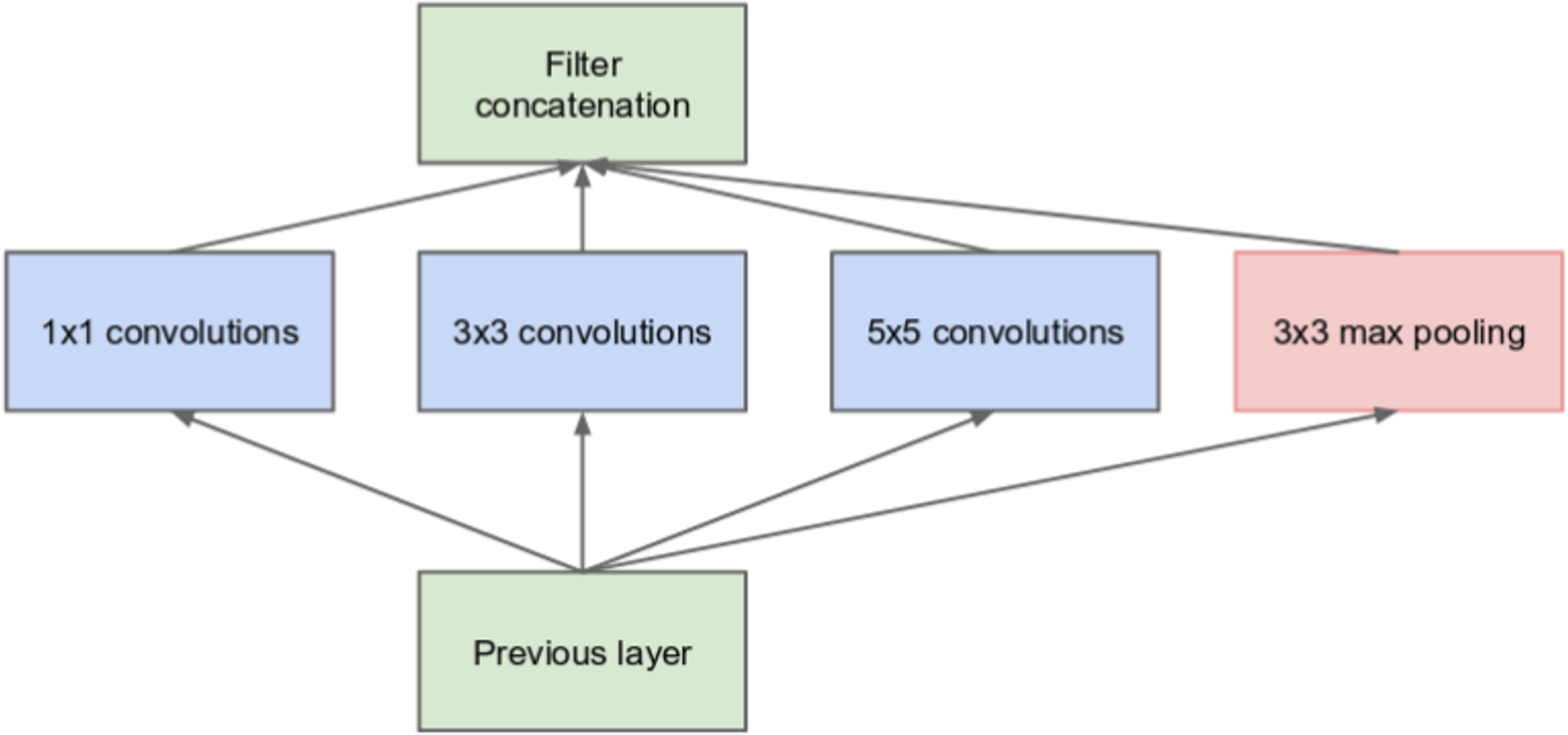
Algunos Modelos: ResNet





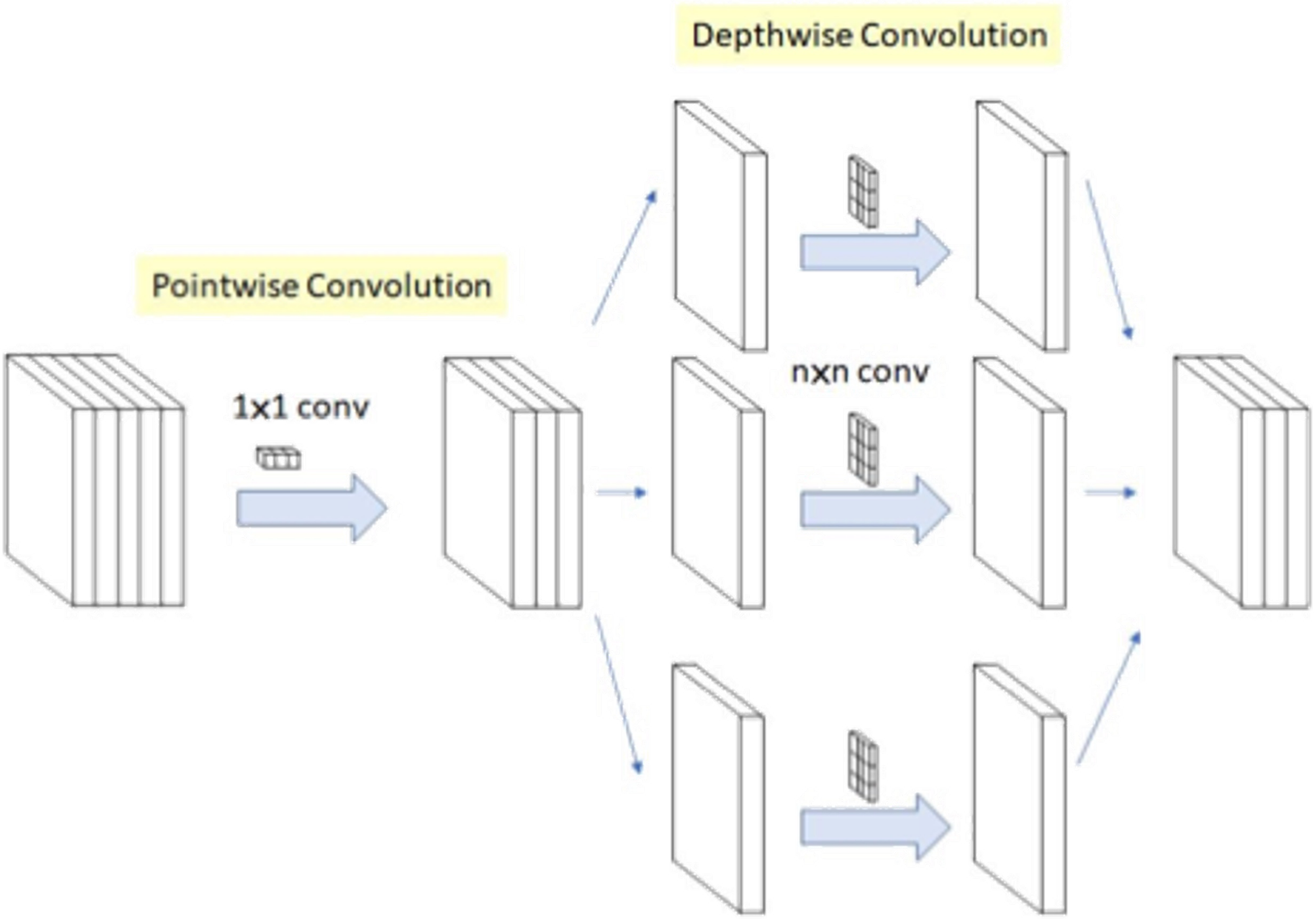
Algunos Modelos: ResNet

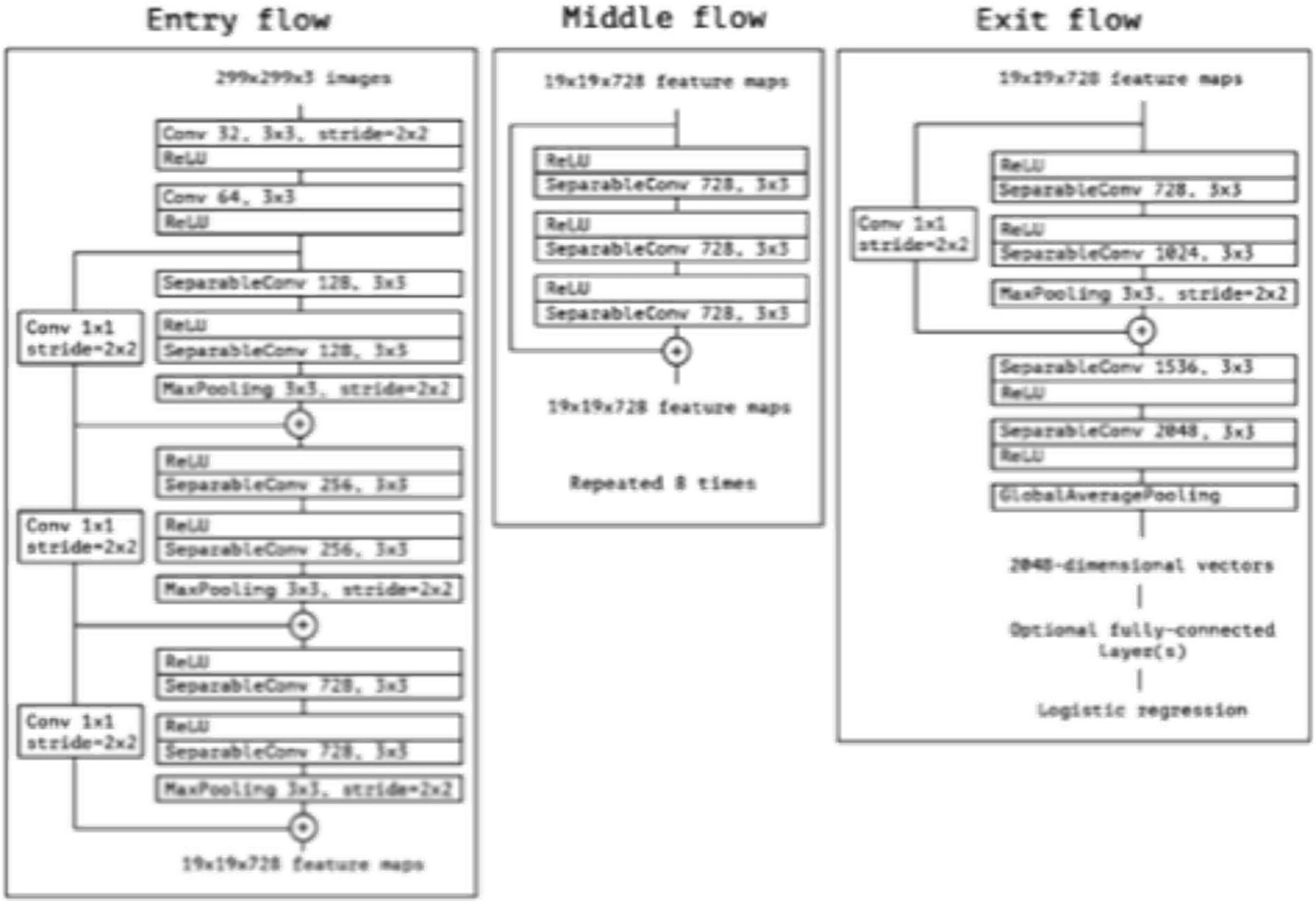
Algunos modelos: Inception



Algunos modelos: Inception

Algunos modelos: Xception




Entry flow
299x299x3 images
Conv 32, 3x3, stride=2x2
ReLU
Conv 64, 3x3
ReLU
SeparableConv 128, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 128, 3x3
MaxPooling 3x3, stride=2x2
ReLU
SeparableConv 256, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 256, 3x3
MaxPooling 3x3, stride=2x2
ReLU
SeparableConv 728, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 728, 3x3
MaxPooling 3x3, stride=2x2
19x19x728 feature maps
Middle flow
19x19x728 feature maps
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 728, 3x3
19x19x728 feature maps
Repeated 8 times
Exit flow
19x19x728 feature maps
ReLU
SeparableConv 728, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 1024, 3x3
MaxPooling 3x3, stride=2x2
SeparableConv 1536, 3x3
ReLU
SeparableConv 2048, 3x3
ReLU
GlobalAveragePooling
2048-dimensional vectors
Optional fully-connected
layer(s)
Logistic regression

¿Dónde consultar modelos existentes?

- Papers with code: https://paperswithcode.com
- Tensorflow Hub: https://www.tensorflow.org/resources/models-datasets
- Model Zoo: https://modelzoo.co
- Hugging face: https://huggingface.co/models

Neural Style Transfer

deeplearning.ai

Neural style transfer


Content Style

Generated image


Content Style

Generated image

[Images generated by Justin Johnson]

Neural style transfer cost function

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: $100 \times 100 \times 3$

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

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

Content cost function

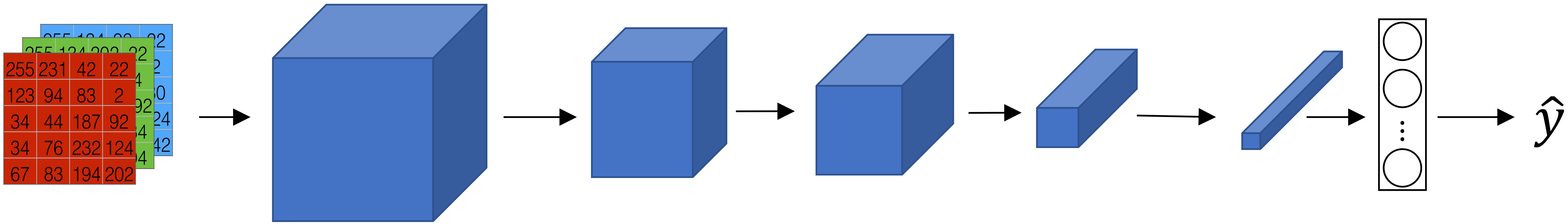
$$J(G) = \underset{\uparrow \atop \gamma}{\alpha} J_{content}(C, G) + \beta\, J_{style}(S, G)$$

- 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

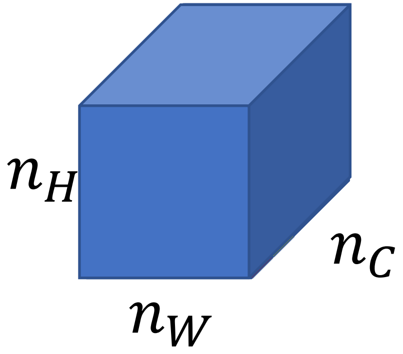
$$J_{content}(C, G) = \frac{1}{2} \underbrace{\| a^{[l](C)} - a^{[l](G)} \|^2}$$

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

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.

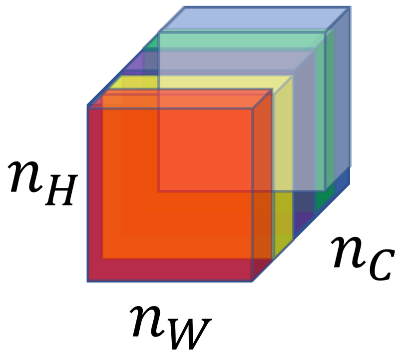


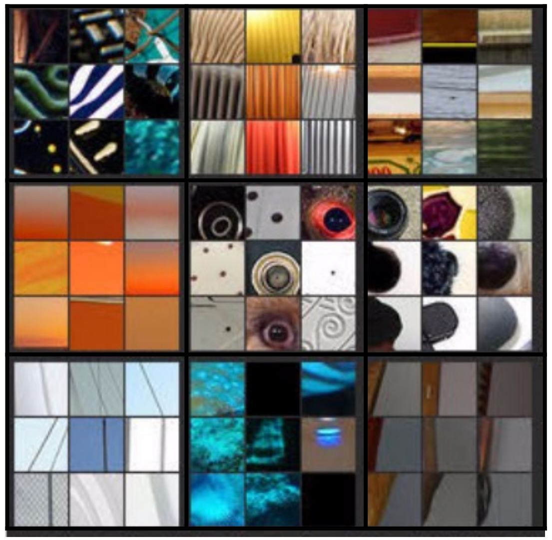
How correlated are the activations across different channels?

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

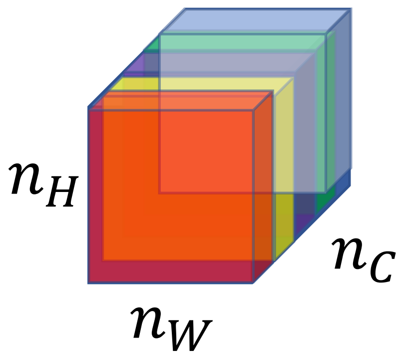
Intuition about style of an image

Style image





Generated Image



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

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

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

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

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

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

"Gram matrix"

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

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

β

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

Style cost function

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

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

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

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

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

Ejemplos

Jose Luís Aguilera Luzanía

Ejemplos

Christopher Arce Díaz

Ejemplos

Raúl Murcia Yocupicio

Ejemplos

Ximena Sandoval