Transferencia de estilo

(Neural Style Transfer)

Agenda

01
Redes convolucionales (CNN)

02
Activaciones en una red convolucional

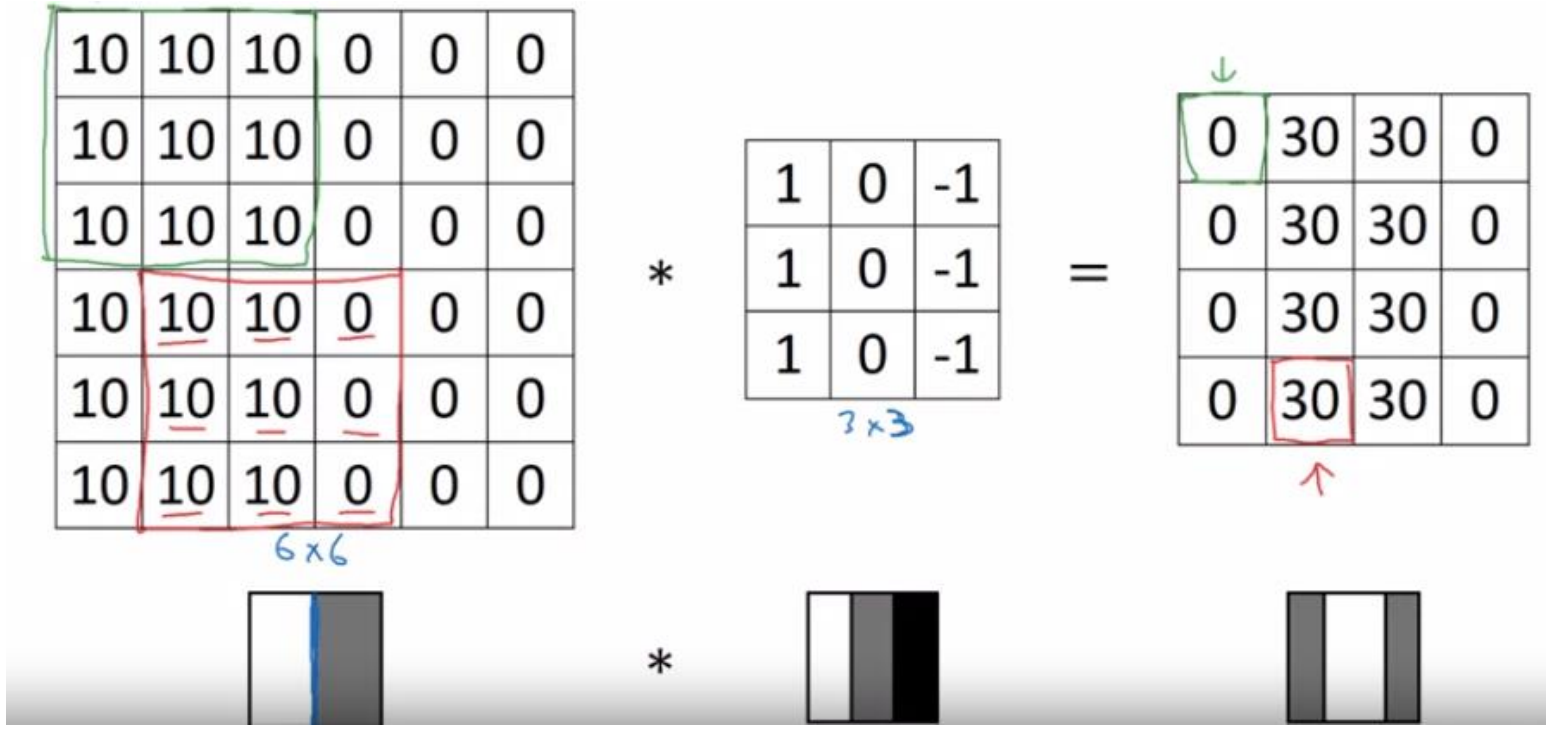
03
Objetivo de transferencia de estilo (NST)

04
Problema "tradicional" versus NST

05
Función de costo

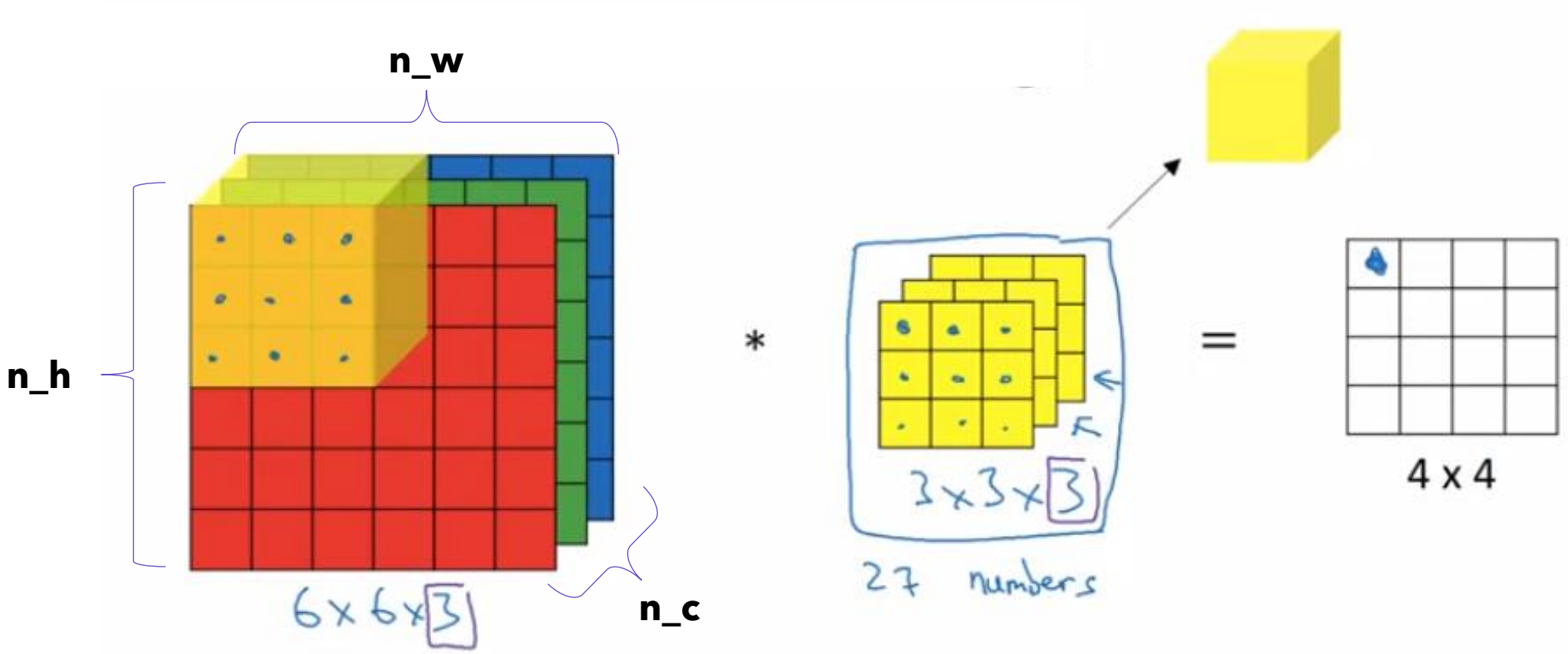
06
Ejemplo

Filtros de convolución



Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

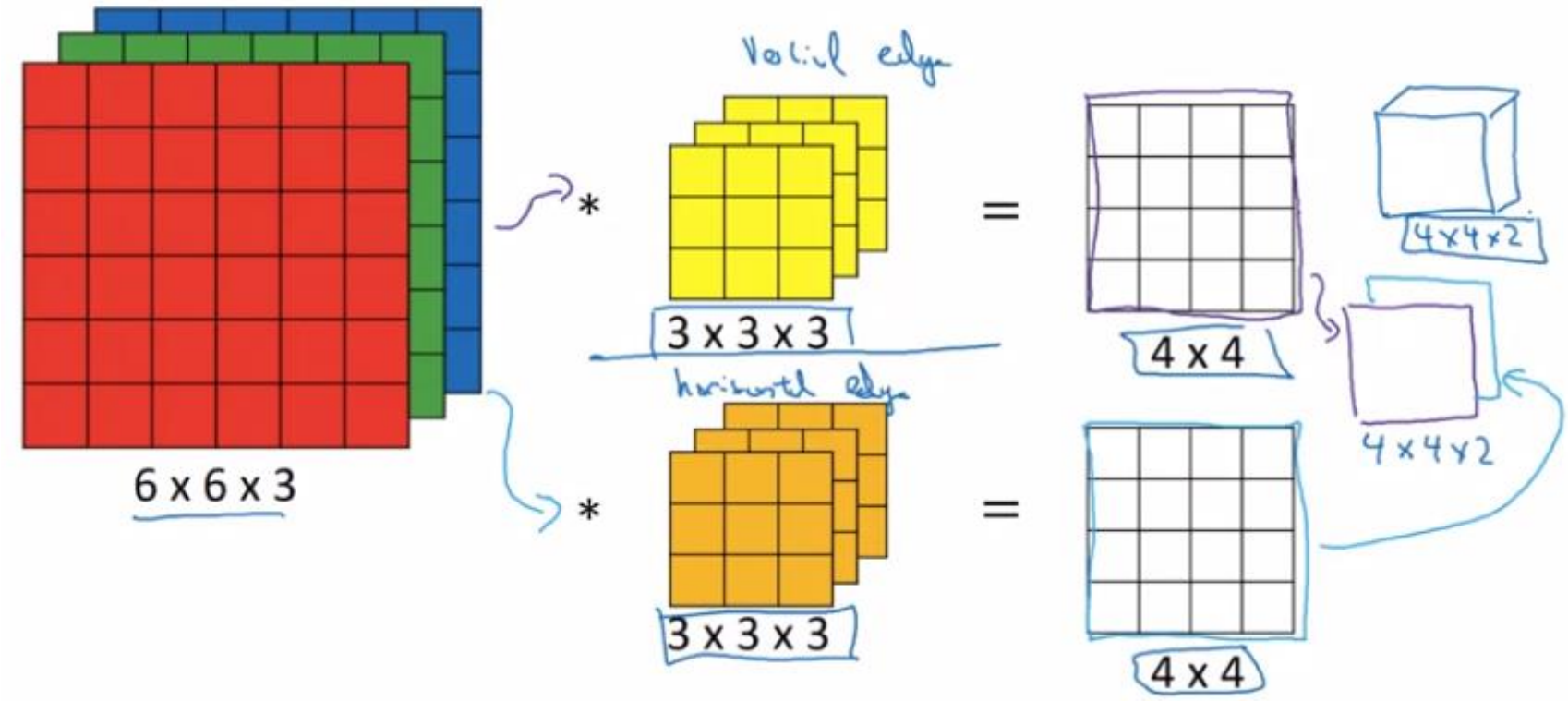
Convoluciones en 3D



Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

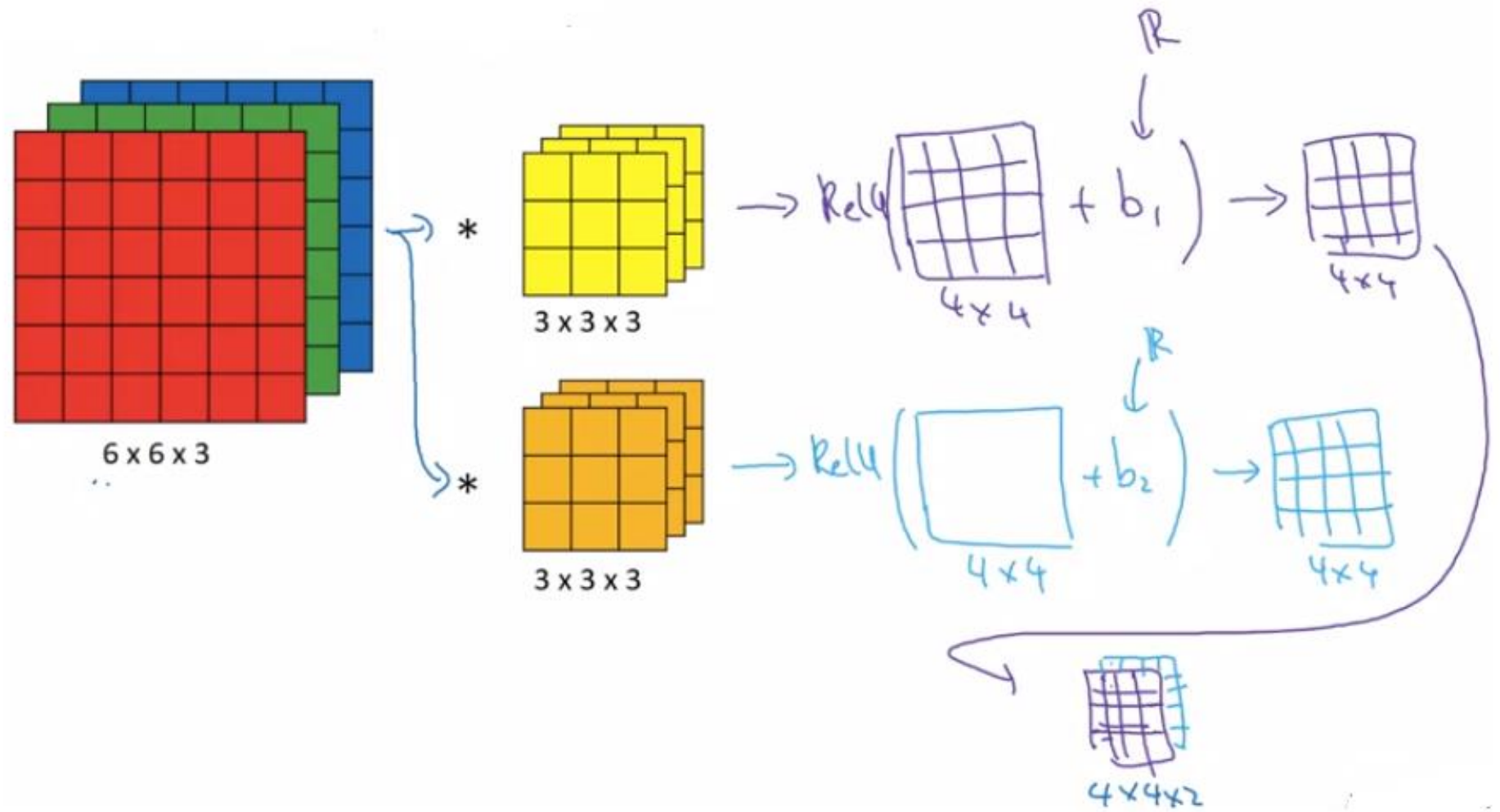
Aplicación de múltiples filtros

- Es posible aplicar varios (**n_c**) filtros de manera simultánea.
- Esto a su vez produce tensores con nueva profundidad



Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Activación de una capa convolucional

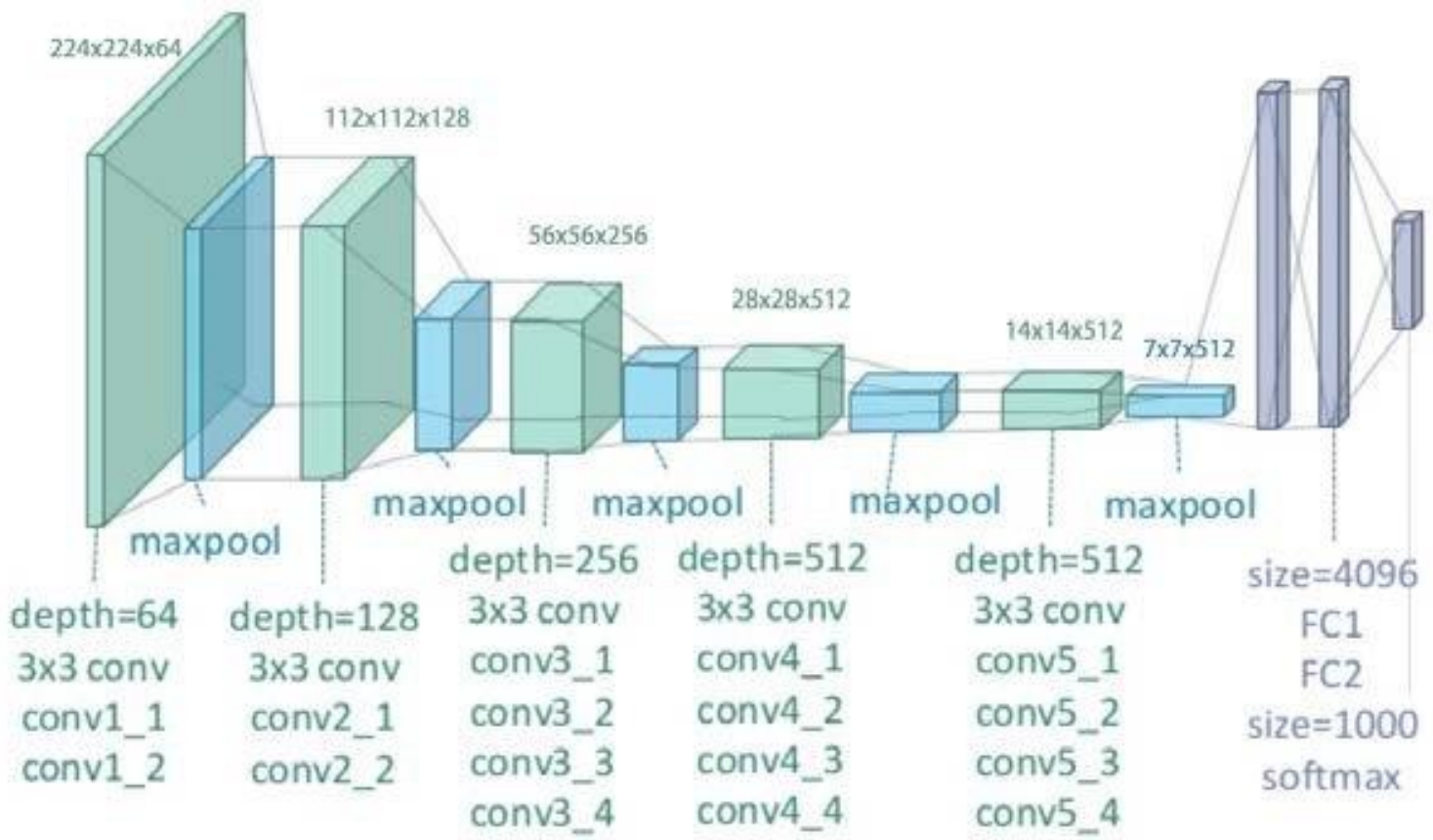


Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

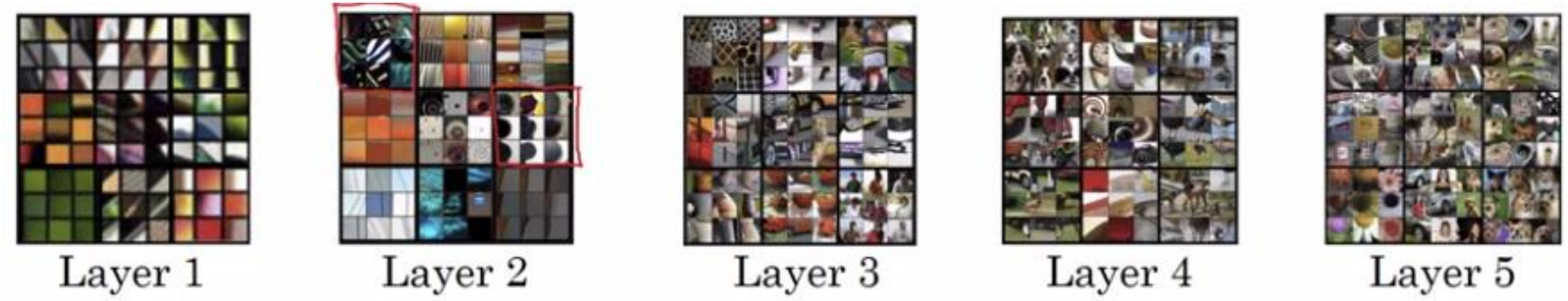
Arquitecturas de CNN

- Combinaciones de capas convolucionales + *pooling*
- Una o más capas densas (FC) al final
- Generalmente, a medida que se añaden más capas:
  - **n_h** y **n_w** 
  - **n_c** 

Ejemplo: Arquitectura VGG-19



Tomada de researchgate.



Activaciones de una CNN

Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Capa 1:

Activaciones de una CNN

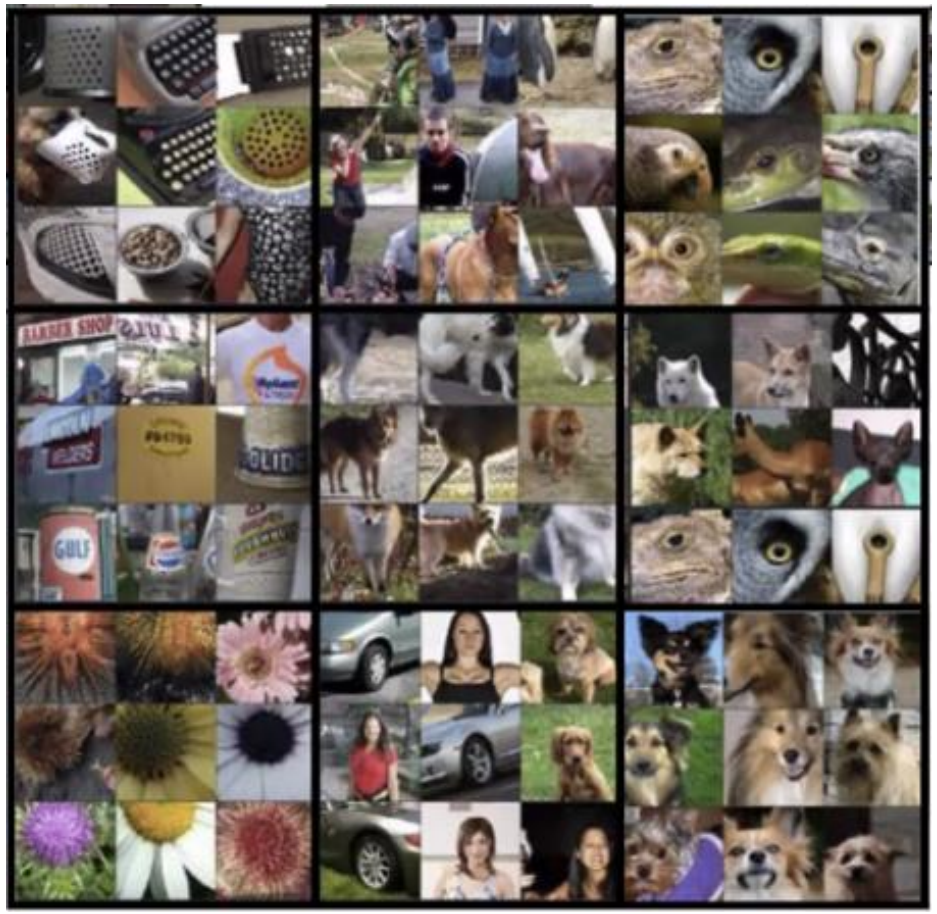
Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Capa 3:

Activaciones de una CNN

Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Capa 5:

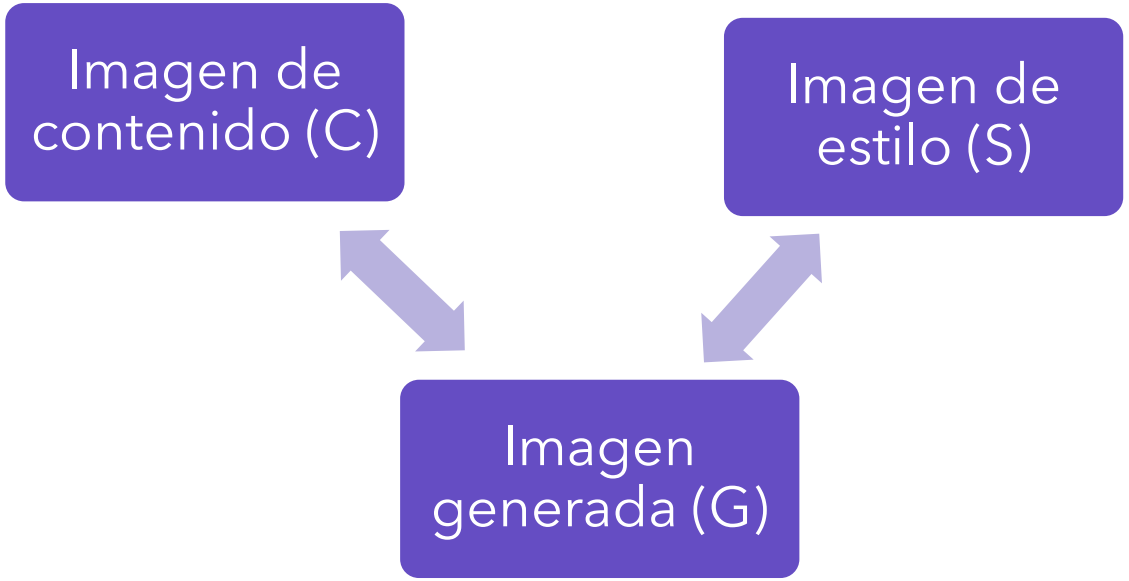


Activaciones de una CNN

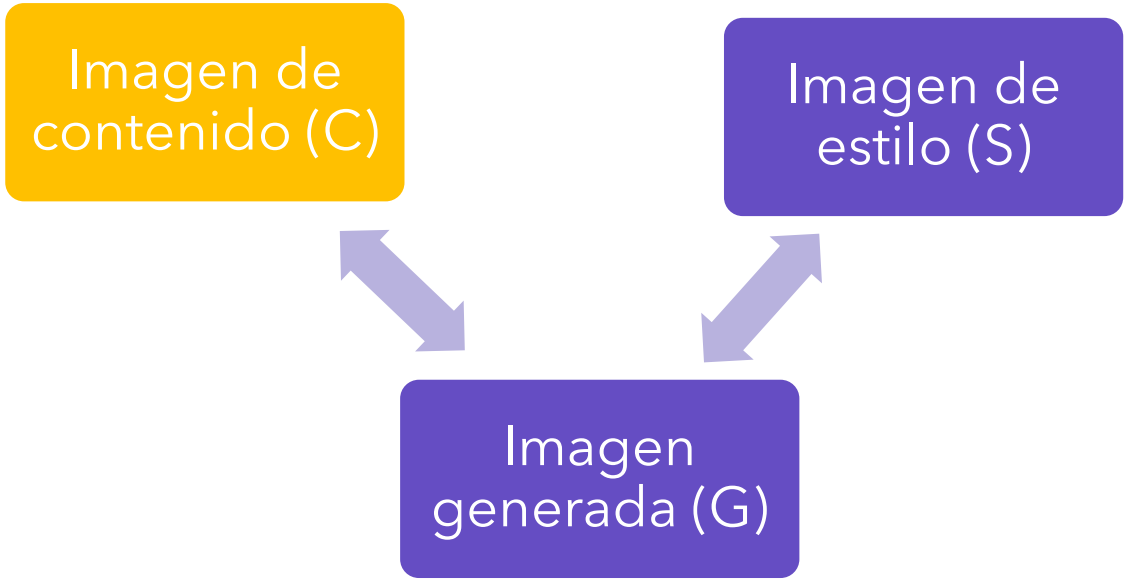
Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Transferencia de estilo (NST)

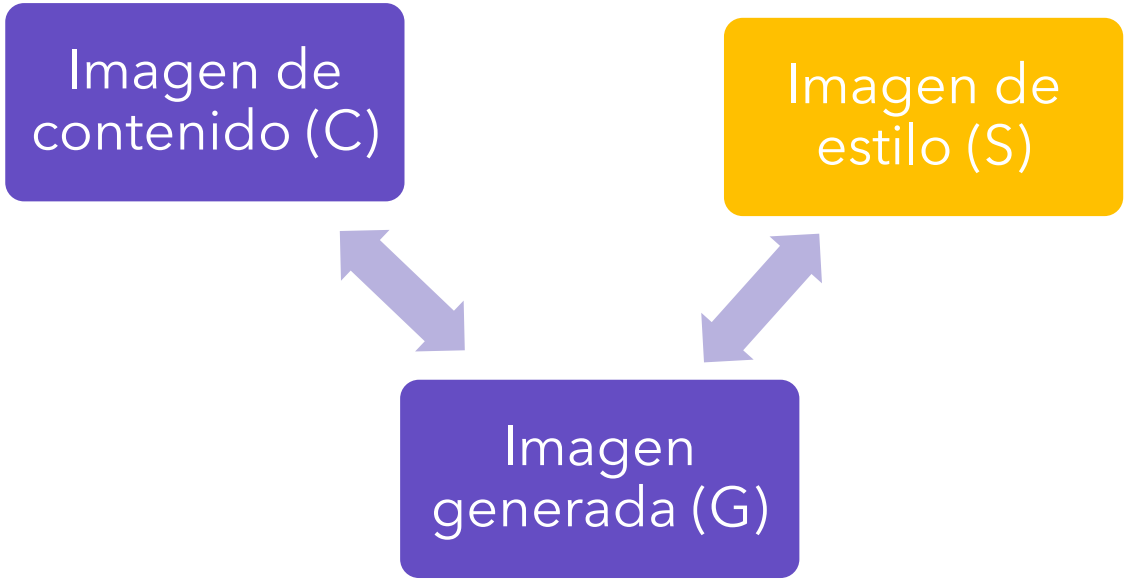
- Algoritmo para generar una nueva imagen a partir de dos imágenes previamente seleccionadas
- L. Gatys, A. Ecker, M. Bethge. A Neural Algorithm of Artistic Style (2015)



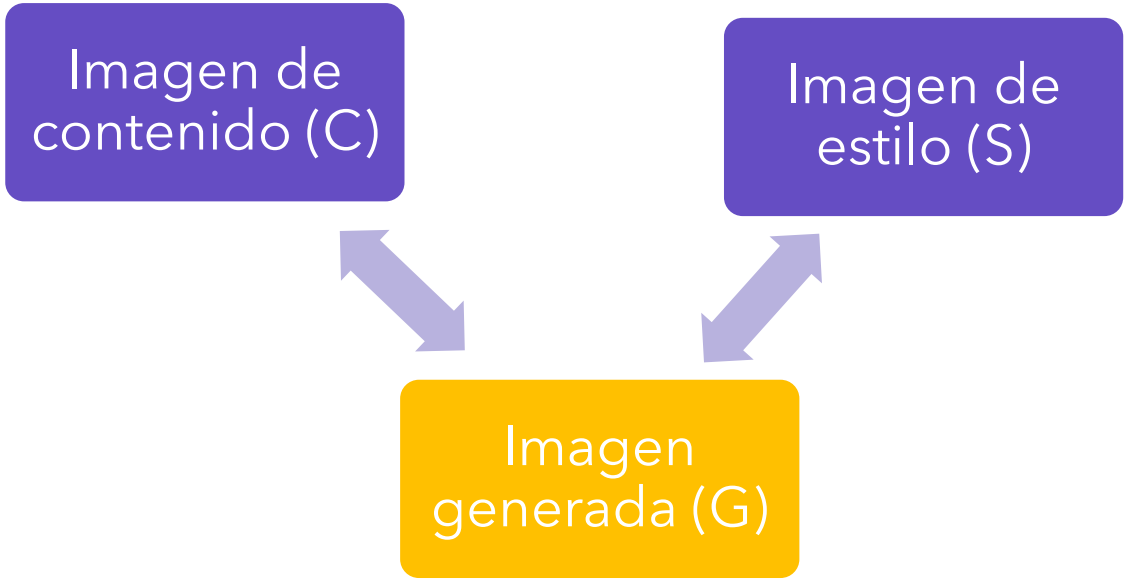
Transferencia de estilo (NST)



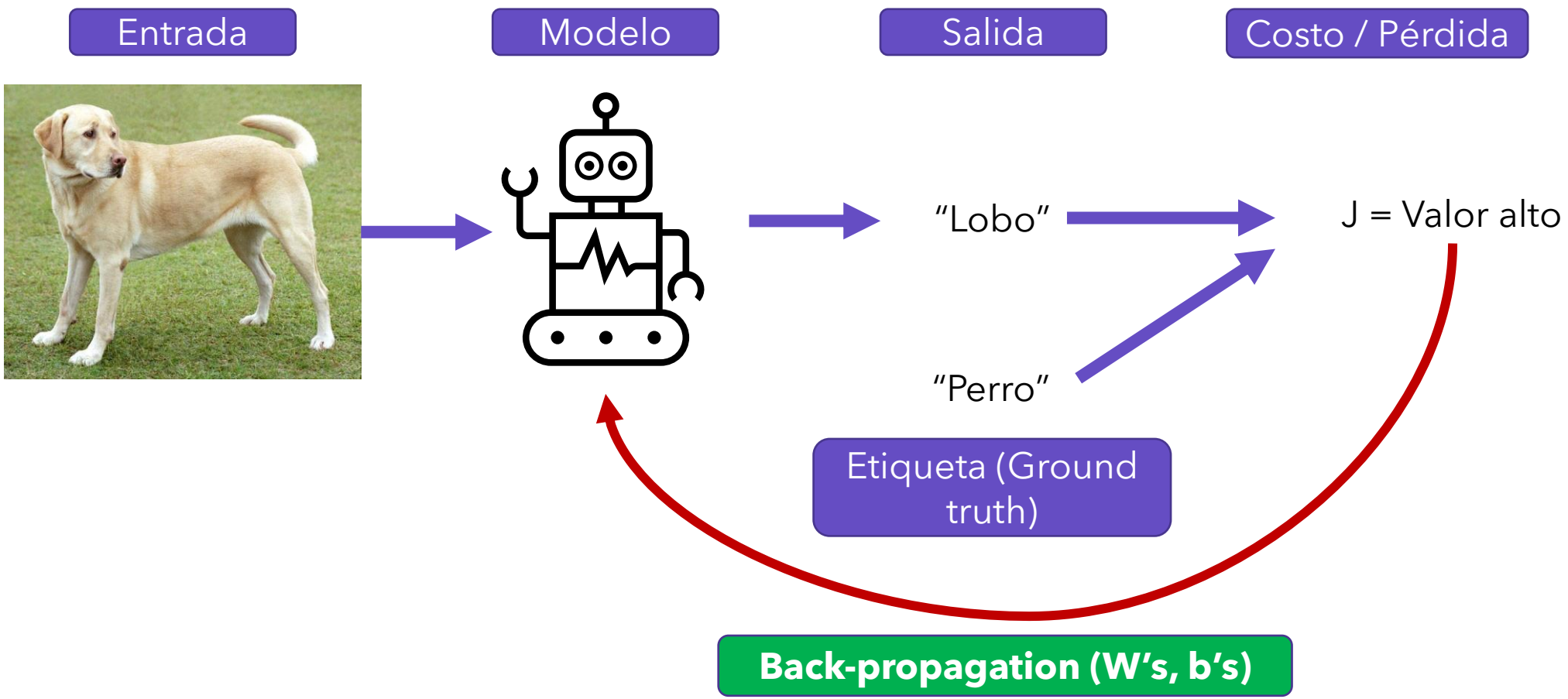
Transferencia de estilo (NST)



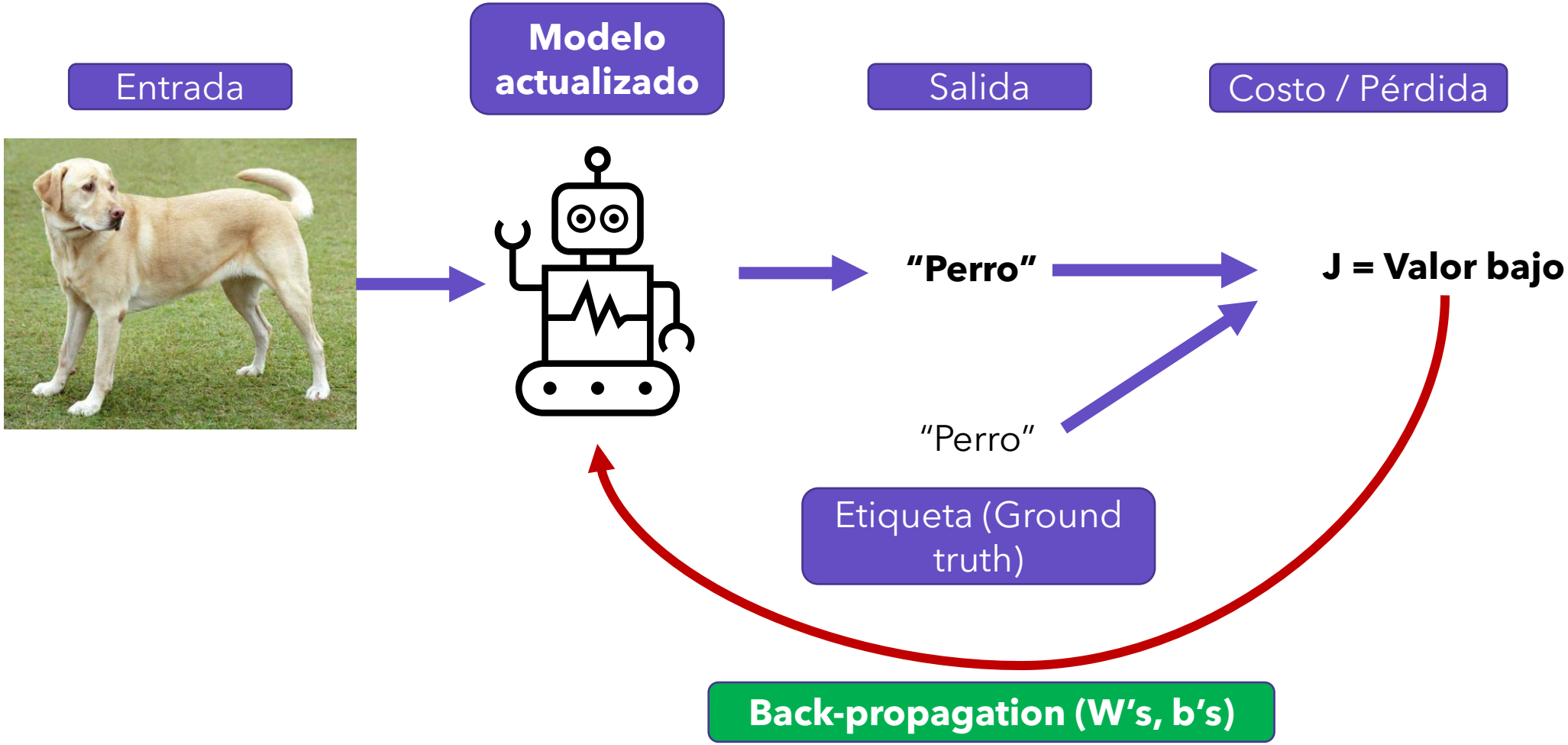
Transferencia de estilo (NST)



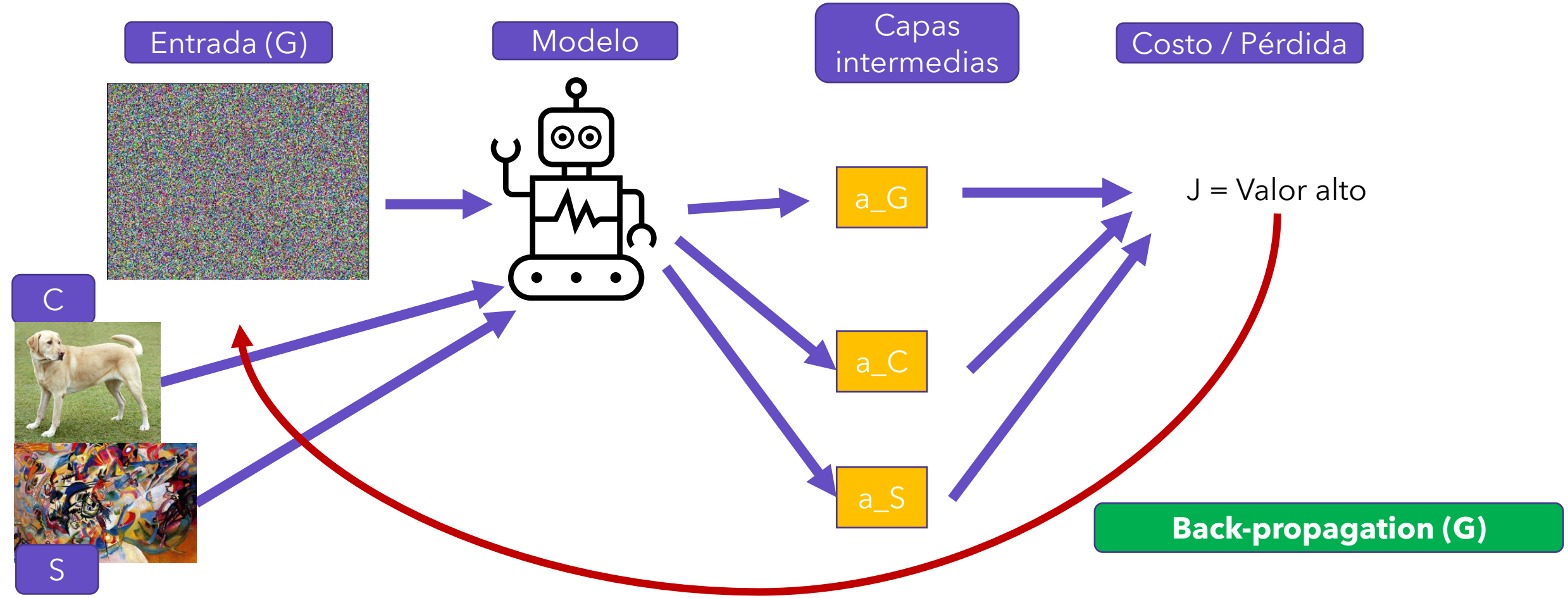
Problema "tradicional" versus NST



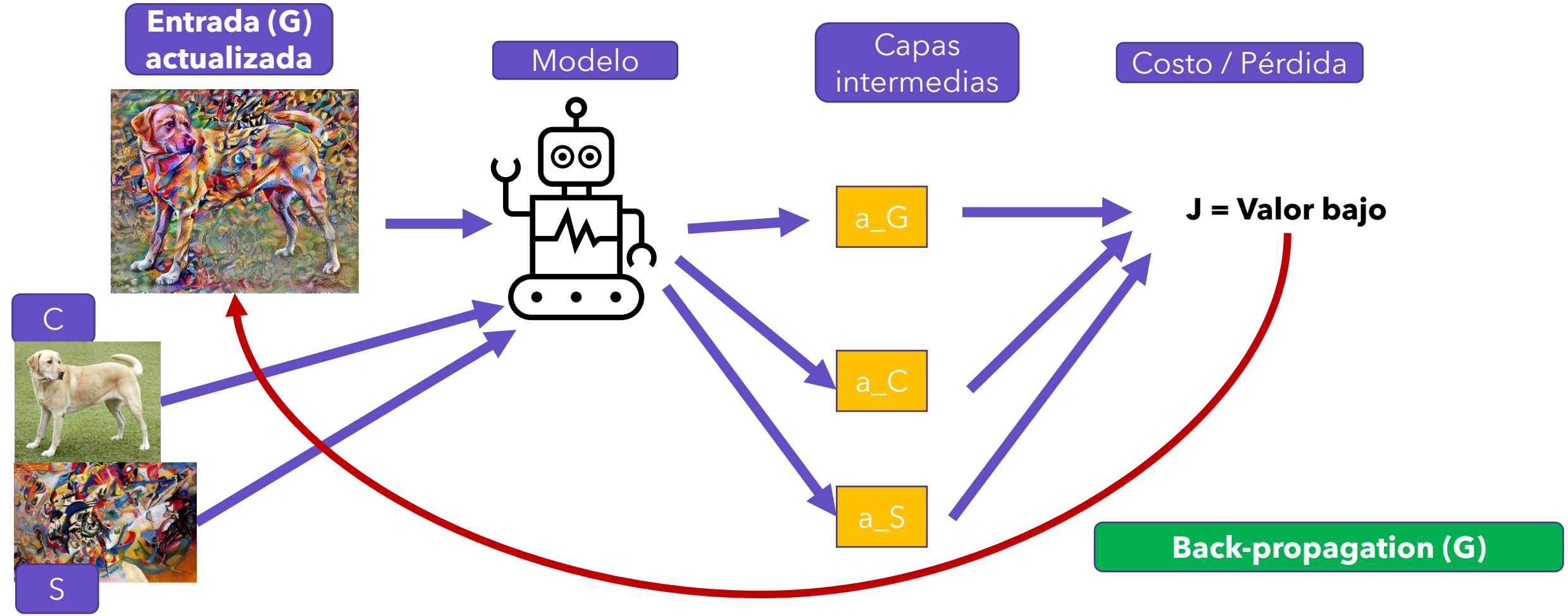
Problema "tradicional" versus NST



Problema “tradicional” versus NST



Problema “tradicional” versus NST



Función de costo para NST

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

Función de costo del contenido

Mide la norma de la diferencia entre la activación de la red en una capa determinada, tanto para la imagen de contenido como para la imagen generada.

$$J_{content}(C, G) = \frac{1}{4 \times n_H \times n_W \times n_C} \sum_{\text{all entries}} (a^{(C)} - a^{(G)})^2$$

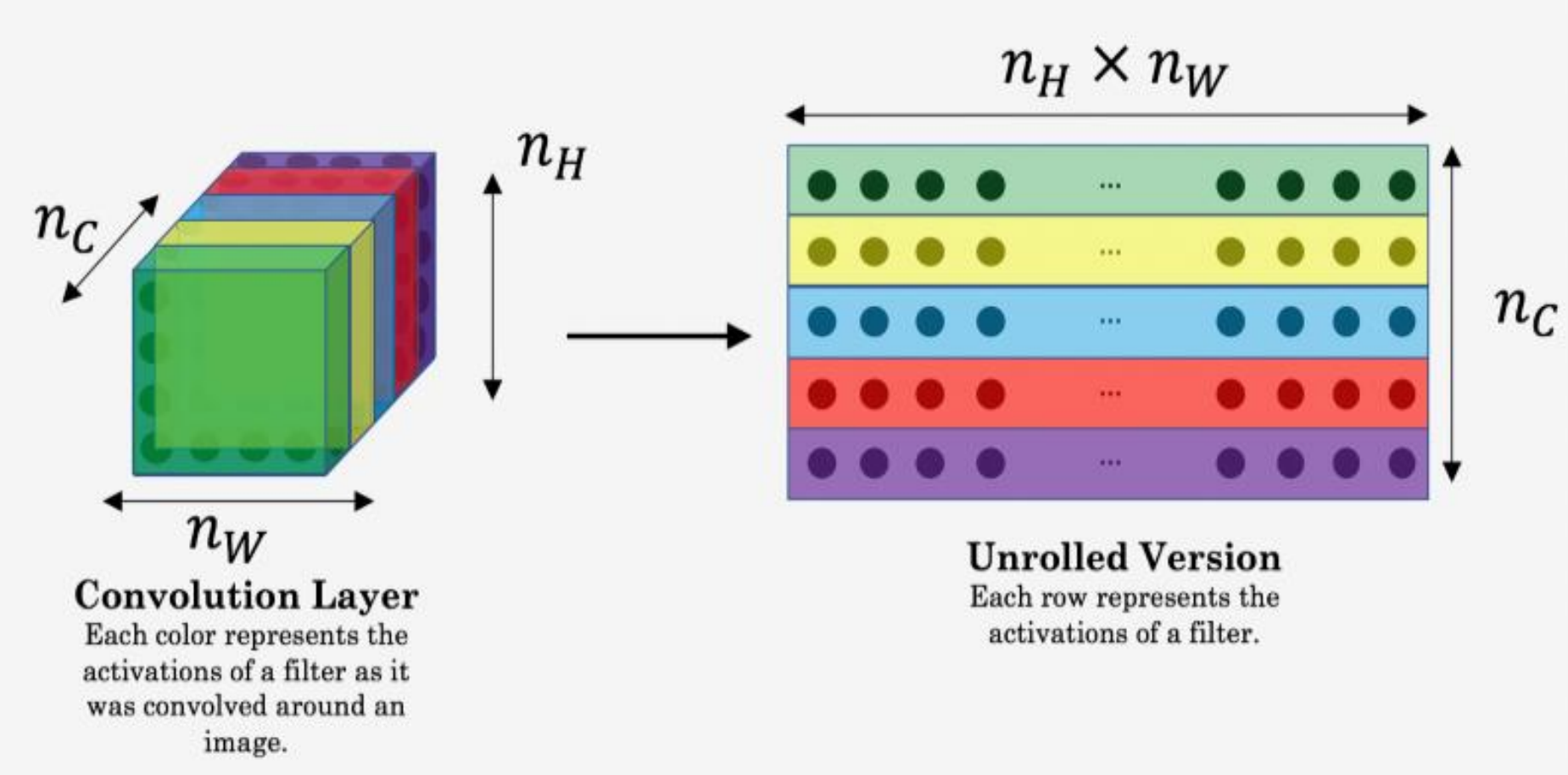
Función de costo del estilo

Mide qué tanto se parecen los estilos entre la imagen de estilo y la imagen generada.

$$J_{style}^{[l]}(S,G) = \frac{1}{4 \times {n_C}^2 \times (n_H \times n_W)^2} \sum_{i=1}^{n_C} \sum_{j=1}^{n_C} (G_{(gram)i,j}^{(S)} - G_{(gram)i,j}^{(G)})^2$$

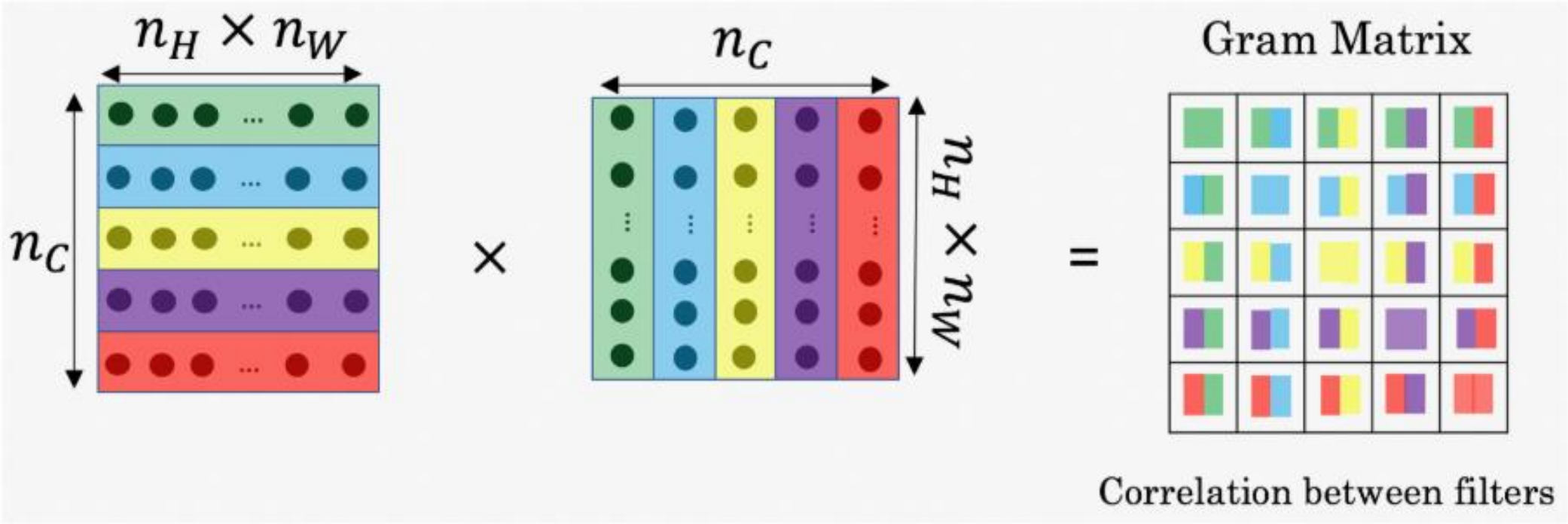
$$J_{style}(S,G) = \sum_{l} \lambda^{[l]} J_{style}^{[l]}(S,G)$$

Cálculo de correlación entre filtros convolucionales



Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Cálculo de correlación entre filtros convolucionales

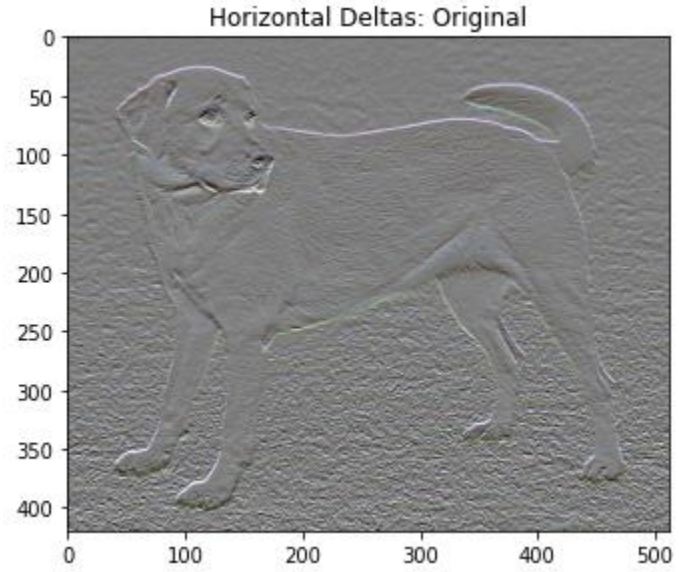


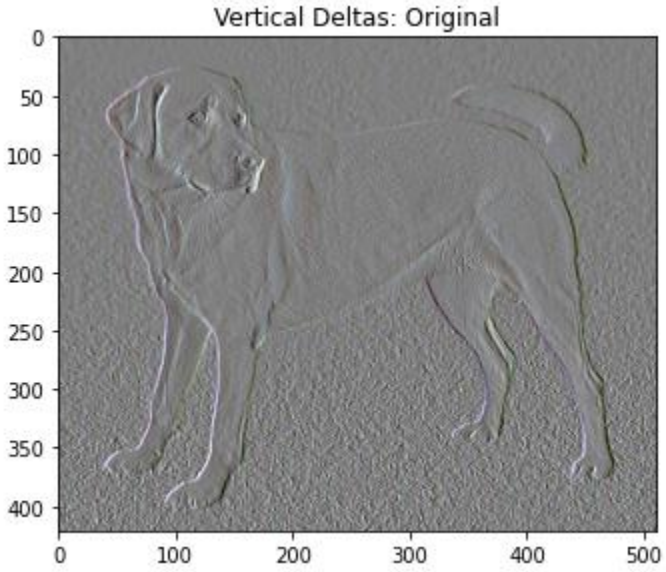
$$\mathbf{G}_{gram} = \mathbf{A}_{unrolled}\mathbf{A}_{unrolled}^{T}$$

Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Función de costo de variación total

Mide diferencias locales entre pixeles (horizontales y verticales). Se minimiza para que la imagen sea coherente localmente.





Referencia: https://www.tensorflow.org/tutorials/generative/style_transfer

Proceso de entrenamiento

1. Initiate G randomly

   G: $100 \times 100 \times 3$

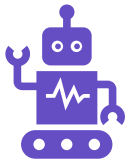
   ↑ RGB

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

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

Referencia: Curso Andrew NG https://www.coursera.org/learn/convolutional-neural-networks

Hiperparámetros del entrenamiento

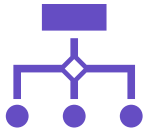


Modelo pre entrenado a utilizar

Pesos de los costos (alpha, beta)

Capa en la que se calcula el costo de contenido

Capa(s) en la(s) que se calcula el costo de estilo, y sus lambdas



Optimizdor (tipo, learning rate, otros parámetros)

Ejemplo de uso