Redes Neuronales Convolucionales

Aprendizaje Automático Aplicado

Agenda

- Un poco de historia
- Aplicaciones
- Motivación de las CNNs
- Convolución en imágenes
- Capa de convolución
- Arquitecturas
- Transfer Learning y Fine Tuning

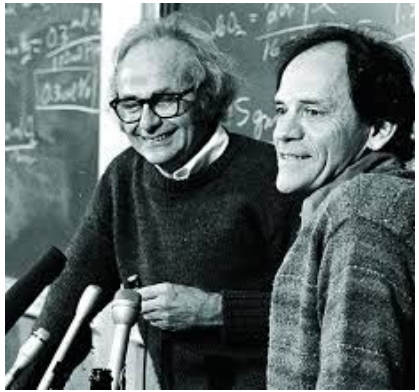
Un poco de historia

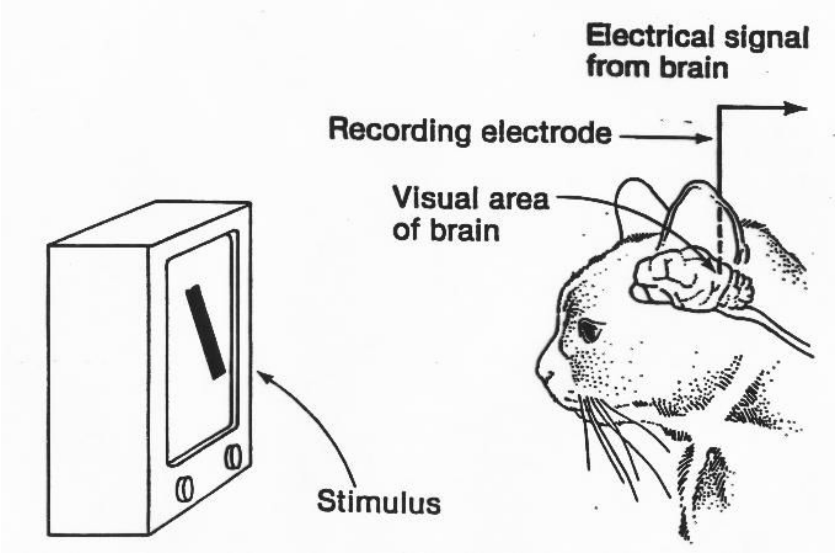
RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT'S STRIATE CORTEX

By D. H. HUBEL* and T. N. WIESEL*

From the Wilmer Institute, The Johns Hopkins Hospital and University, Baltimore, Maryland, U.S.A.

(*Received* 22 *April* 1959)

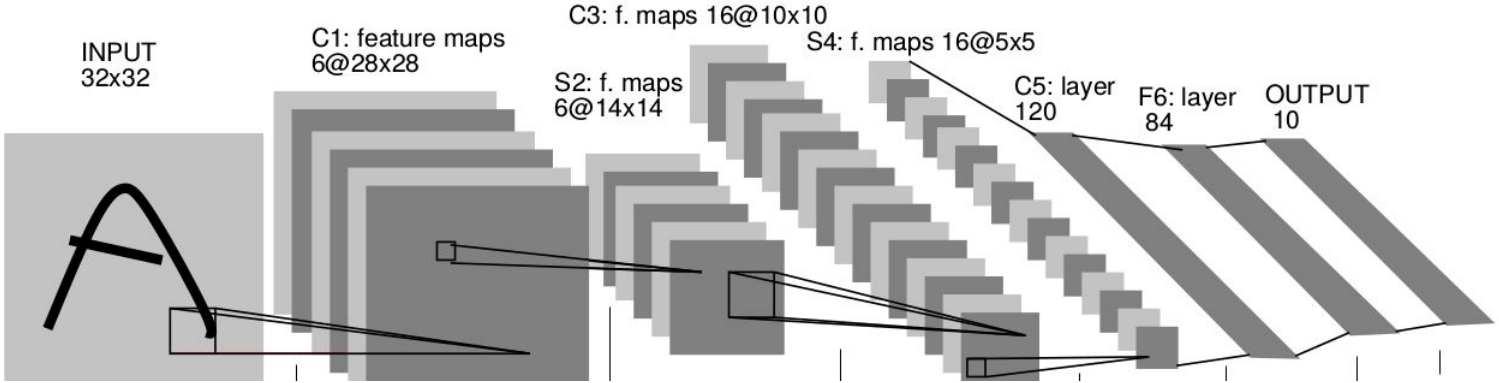




Un poco de historia

- LeNet-5

  Gradient-Based Learning Applied to Document Recognition [Yann LeCun et al., 1998] - (Citado 39.187 veces)



Un poco de historia

- “ImageNet Classification with Deep Convolutional Neural Networks” [Alex Krizhevsky et al., 2012] (citado 86.378 veces)

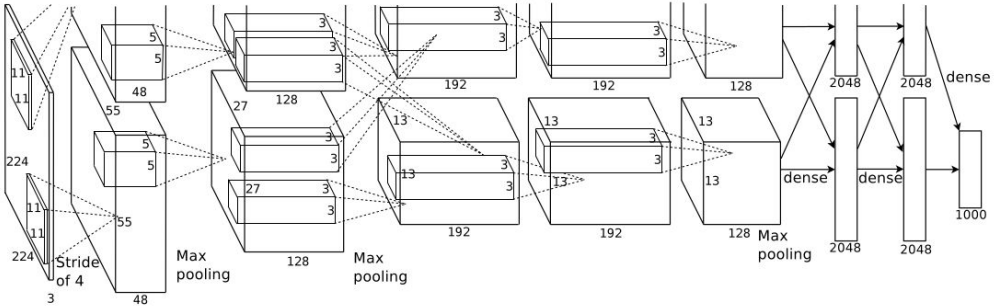
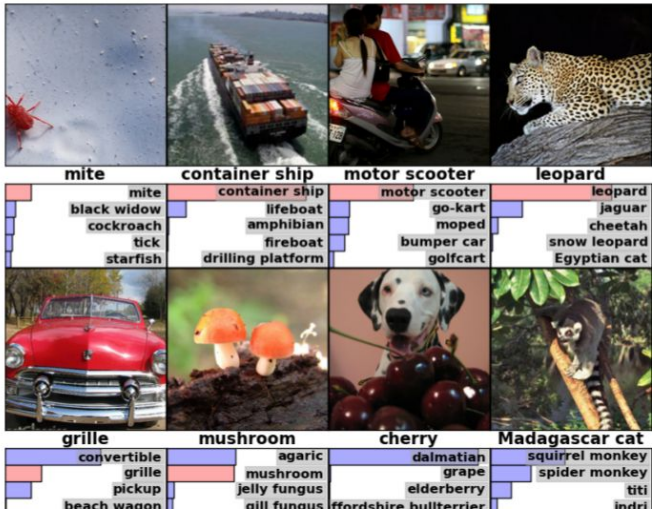
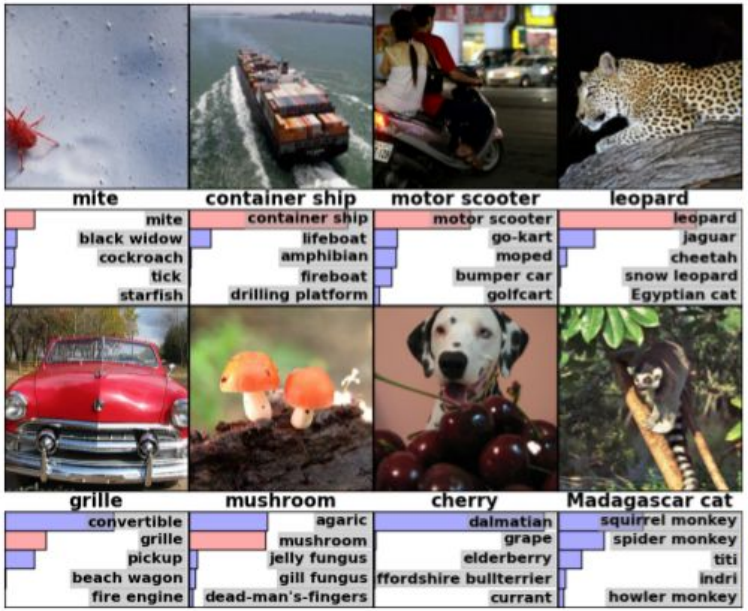


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network’s input is 150,528-dimensional, and the number of neurons in the network’s remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.



Aplicaciones

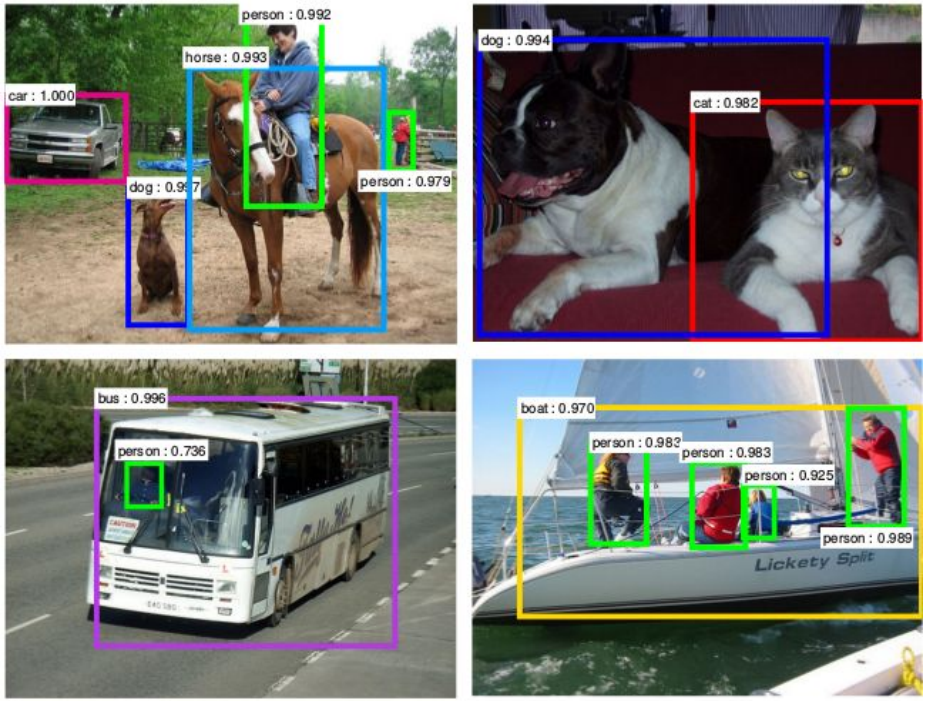
- Clasificación



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012. (Cited by 45305)

Aplicaciones

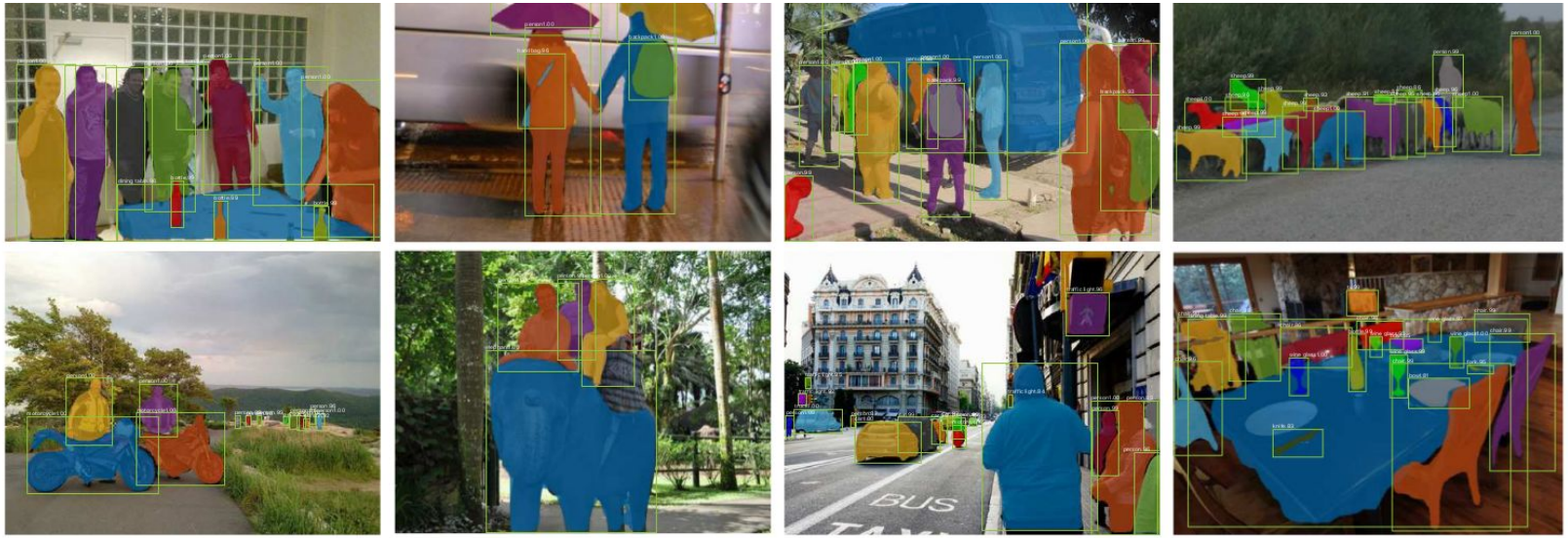
- Detección



Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015. (Cited by 11368)

Aplicaciones

- Segmentación



He, Kaiming, et al. "Mask r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2017. (Cited by 3000)

Aplicaciones

- Descripción de imágenes

man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

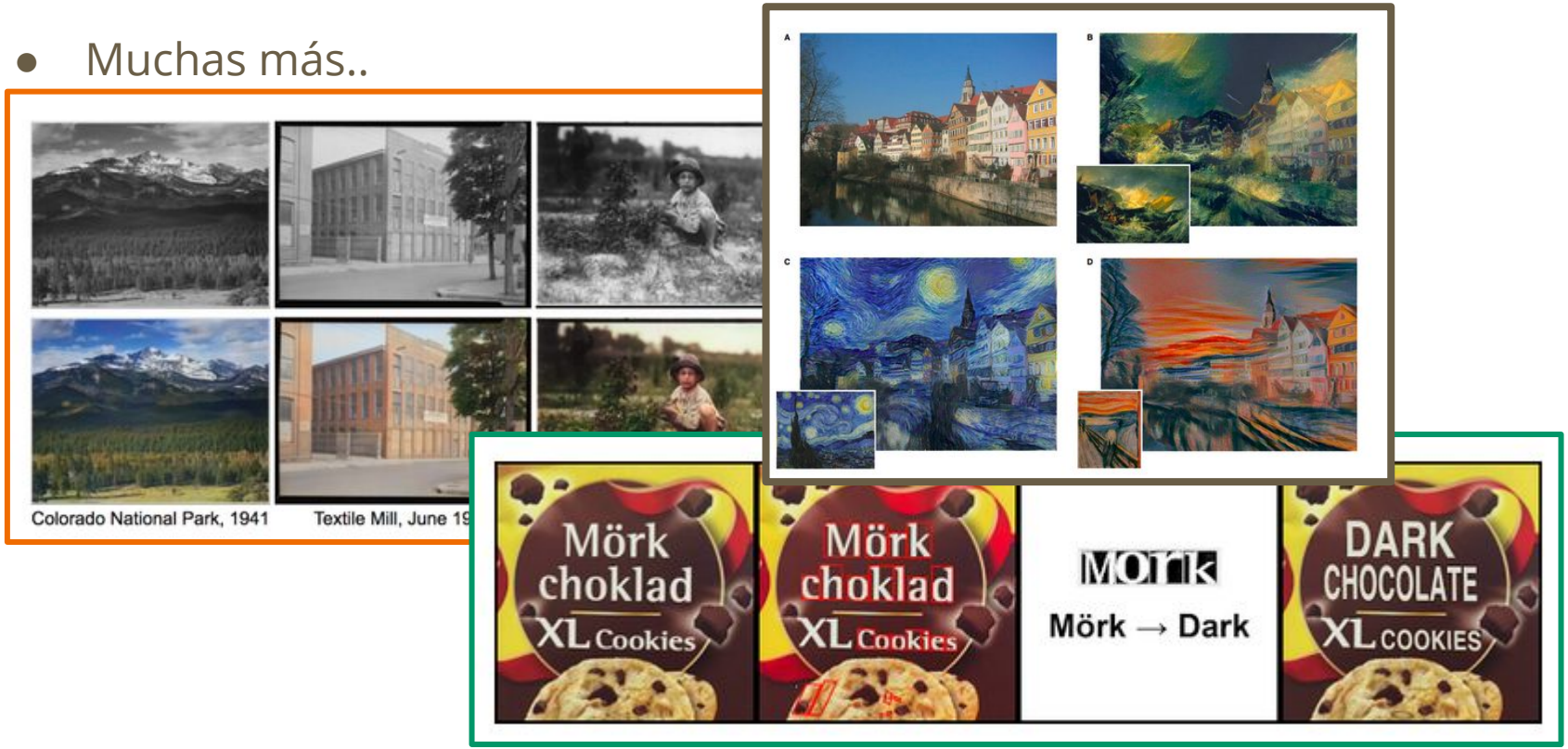
two young girls are playing with lego toy.

boy is doing backflip on wakeboard.

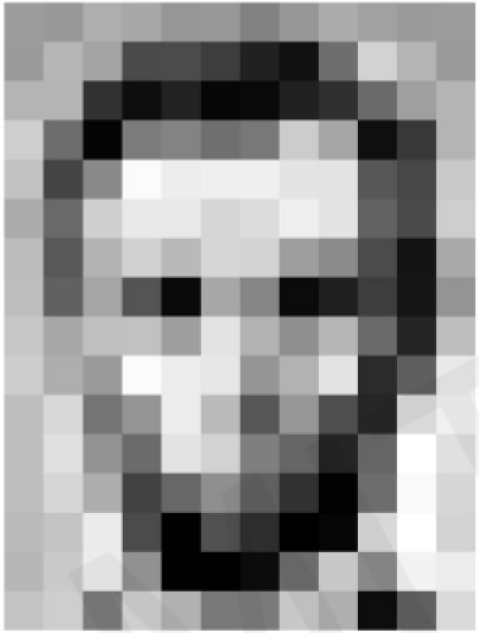
Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. (Cited by 2629)

Aplicaciones

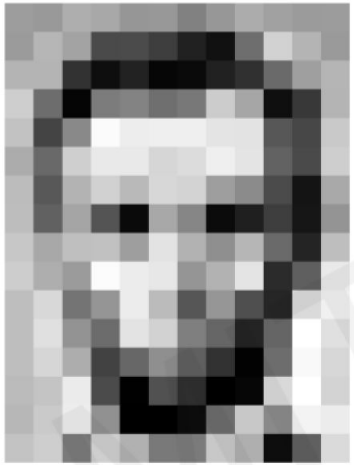
- Muchas más..



Computer vision - qué es una imagen

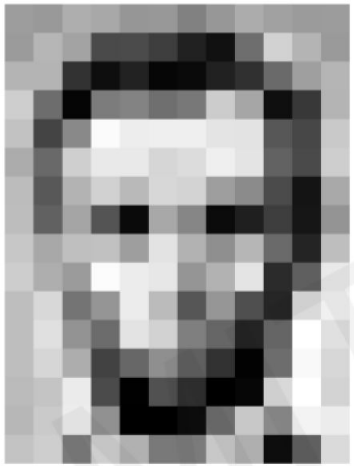


Computer vision - qué es una imagen



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Computer vision - qué es una imagen



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

What the computer sees

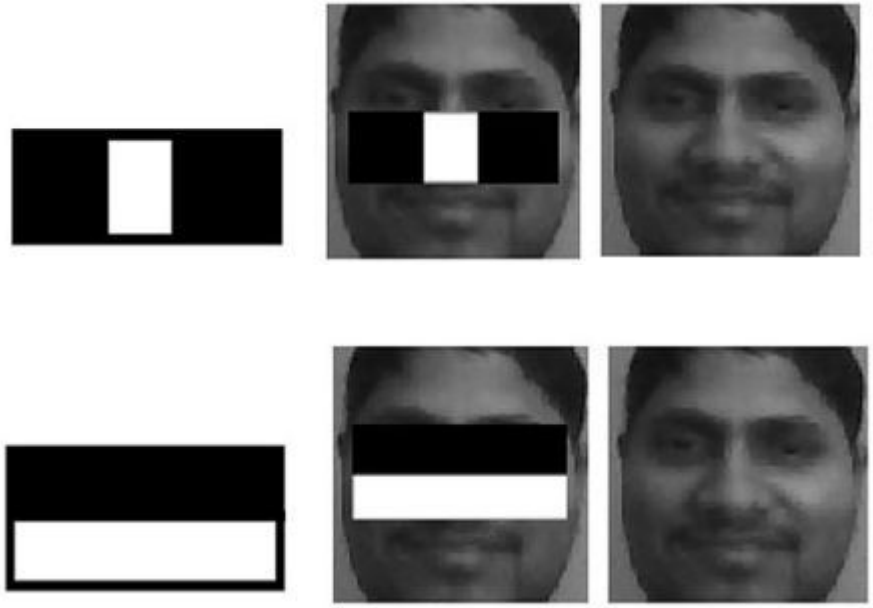
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

Computer vision - features

Computer vision - features manuales

- Se pueden obtener muy buenos resultados, como en el caso de Viola-Jones (2001) para detección de caras.
- Requieren conocimiento del dominio
- Features específicos para cada tipo de objeto.
- No son robustos a traslaciones, rotaciones, espejado, etc.
- Costosos de elaborar.



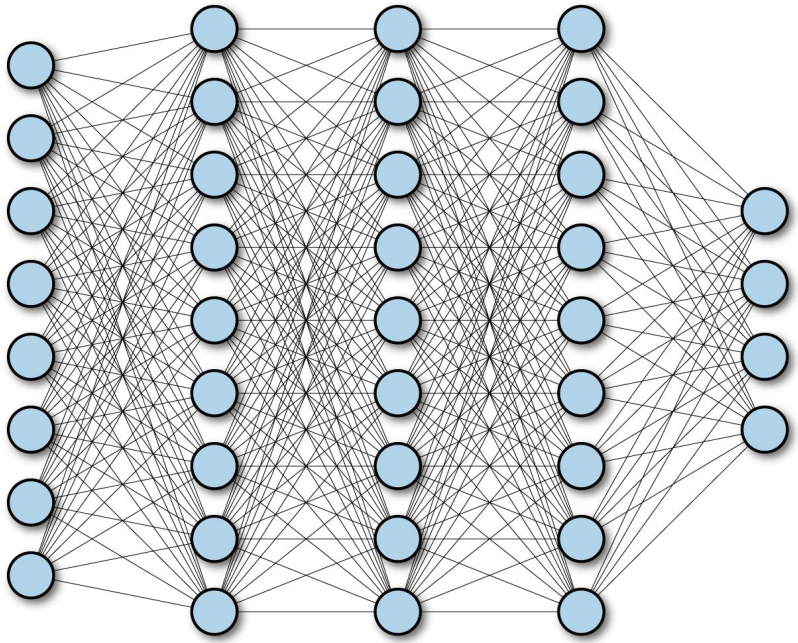
Imágenes tomadas de Wikipedia (https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework)

Computer vision

Computer vision - invariante a escala, rotación, etc

Computer vision - fully connected

- Imágenes a color de 128x128 como entrada:
  - Capa de entrada de dimensión 49k!
- Si además empezamos a apilar capas fully connected, rápidamente se alcanzan cantidades de parámetros difíciles de manejar
  - Demasiados parámetros: sobreajuste
  - Más probable caer en mínimos locales
  - Requiere mucho poder de cómputo



Convolución en imágenes

- Operación lineal entre una imagen y un filtro
- La salida es una nueva imagen
- El valor de cada píxel de la imagen de salida es la suma ponderada de los píxeles de la imagen de entrada y un núcleo de convolución:

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

=

69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

Convolución en imágenes

$$(u * h)(i, j) = \sum_{k,l} u(i-k, j-l)h(k, l)$$

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

*

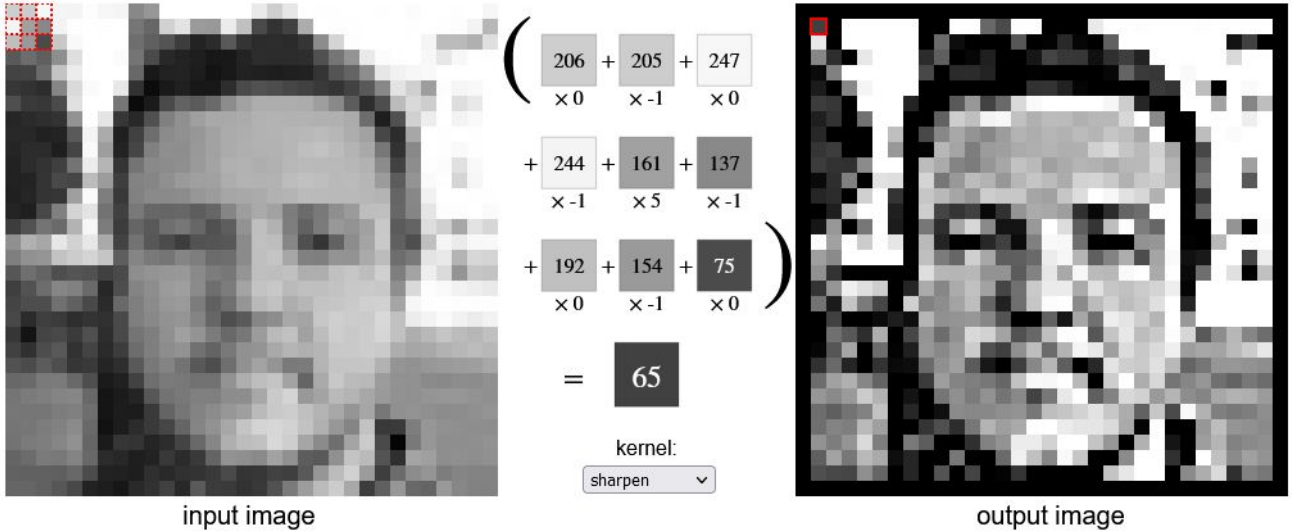
0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

=

69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

Demo online - Image Kernels by Victor Powell

https://setosa.io/ev/image-kernels/



Quiz time!

1. ¿Verdadero o Falso?
    - Diseñar mecanismos para extraer features de imágenes de forma manual es un proceso que requiere cierto conocimiento de dominio pero es simple y barato de replicar para diferentes problemas.
    - Las redes completamente conectadas son mejores para trabajar con imágenes porque capturan todas las posibles relaciones entre píxeles.
    - La utilización de redes convolucionales aplicadas al procesamiento de imágenes ha acelerado sustancialmente el desarrollo del deep learning en general.
2. ¿Cuál es la dimensión de la imagen de salida al convolucionar una imagen de dimensiones (10,10) con un filtro de dimensiones (3,3)?
3. Calcular la matriz resultante:

$$\begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 2 \\ 4 & 8 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Quiz time!

1. ¿Verdadero o Falso?
   - Diseñar mecanismos para extraer features de imágenes de forma manual es un proceso que requiere cierto conocimiento de dominio pero es simple y barato de replicar para diferentes problemas.
   - Las redes completamente conectadas son mejores para trabajar con imágenes porque capturan todas las posibles relaciones entre píxeles.
   - La utilización de redes convolucionales aplicadas al procesamiento de imágenes ha acelerado sustancialmente el desarrollo del deep learning en general.
2. ¿Cuál es la dimensión de la imagen de salida al convolucionar una imagen de dimensiones (10,10) con un filtro de dimensiones (3,3)?
3. Calcular la matriz resultante:

$$\begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 2 \\ 4 & 8 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 10 & 3 \end{bmatrix}$$

Convolución en imágenes

- Aprenden patrones jerárquicos
- Patrones invariantes a traslaciones

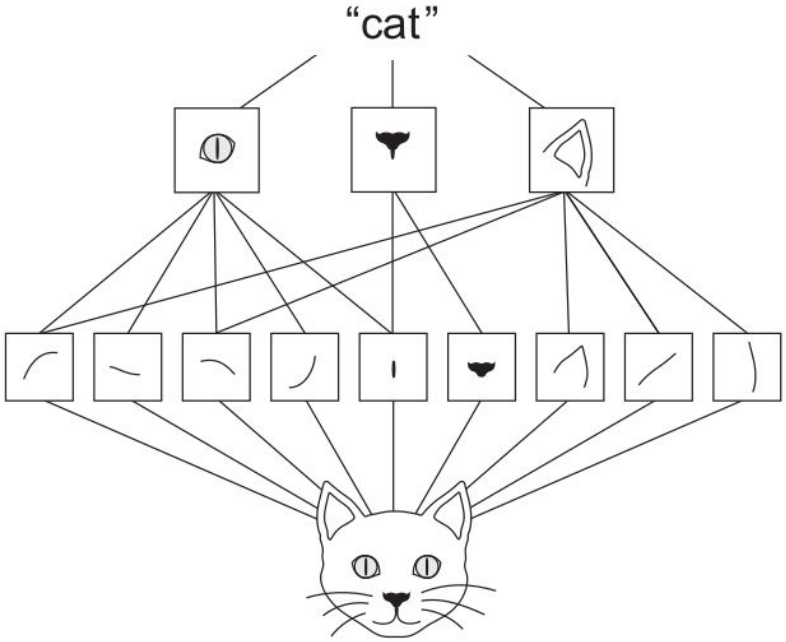
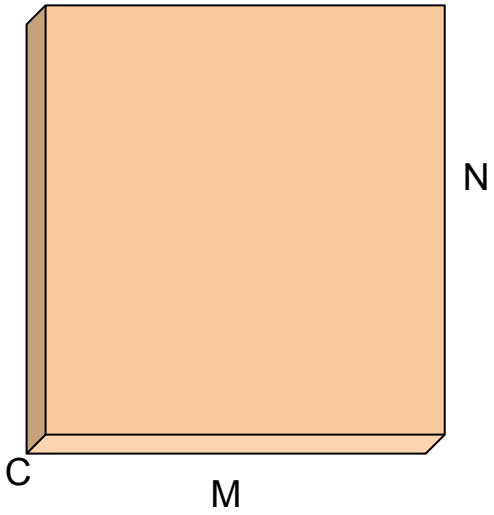
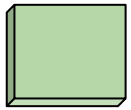


Figure 5.2 The visual world forms a spatial hierarchy of visual modules: hyperlocal edges combine into local objects such as eyes or ears, which combine into high-level concepts such as “cat.”

Capa de convolución

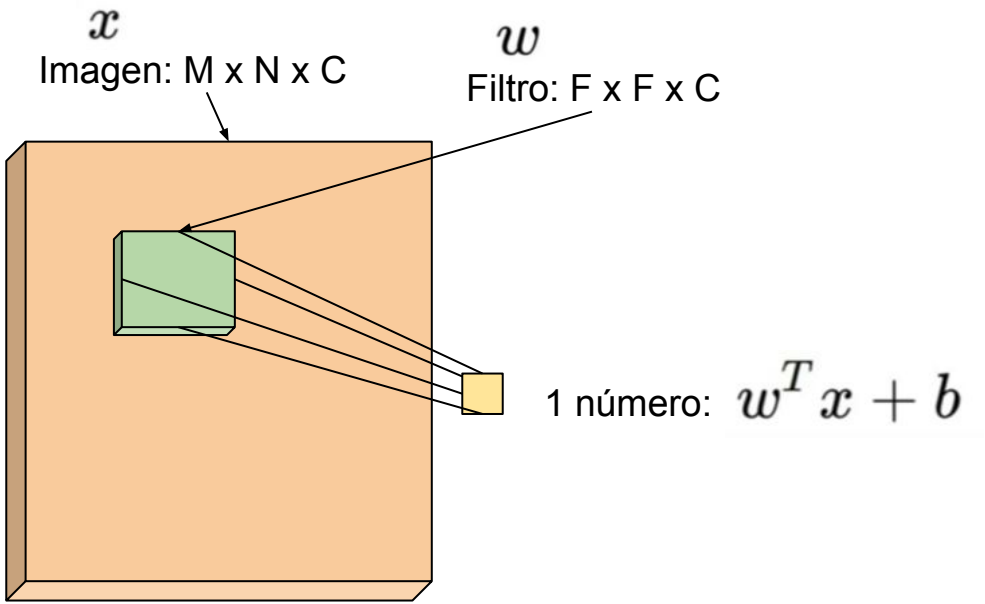
Imagen: M x N x C



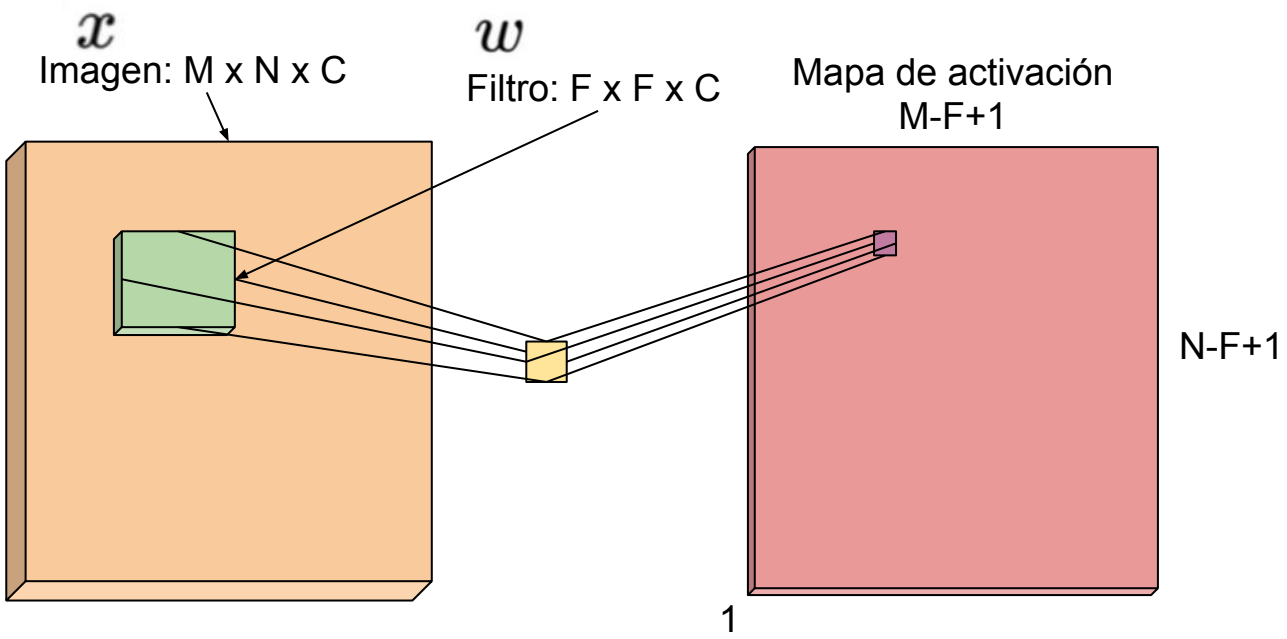


Filtro: F x F x C

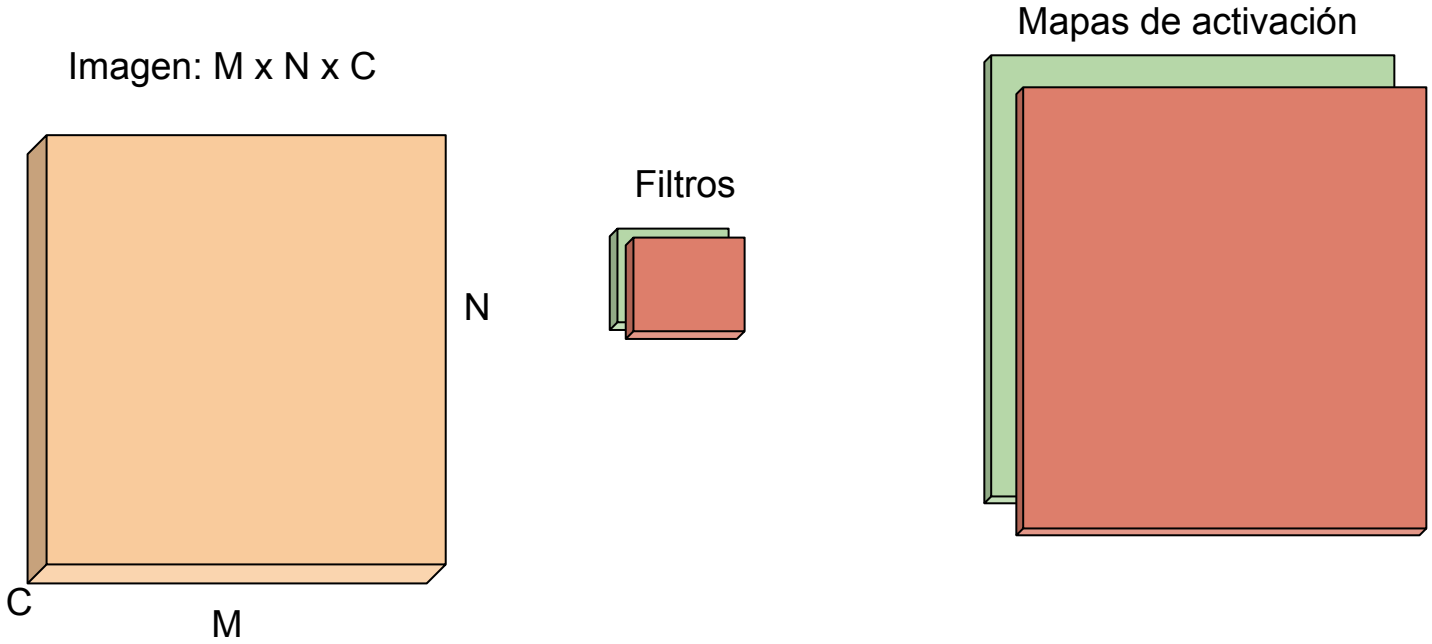
Capa de convolución



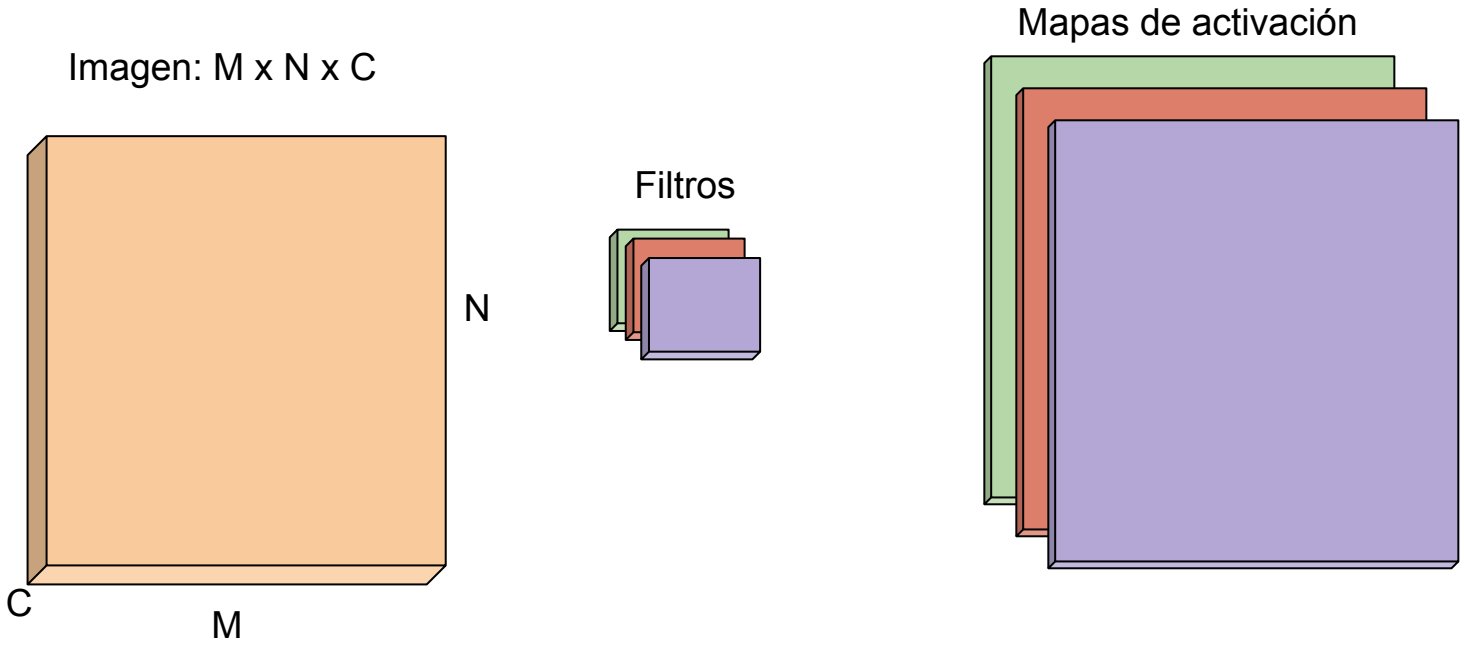
Capa de convolución



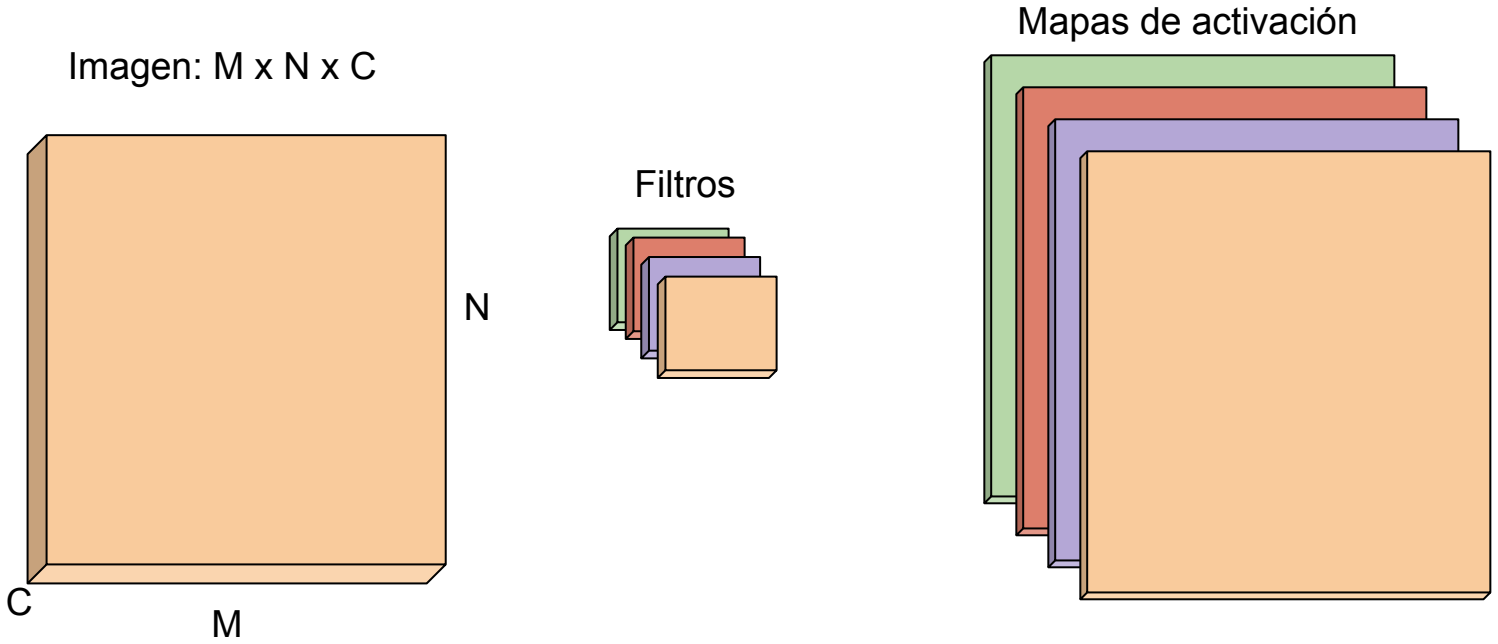
Capa de convolución



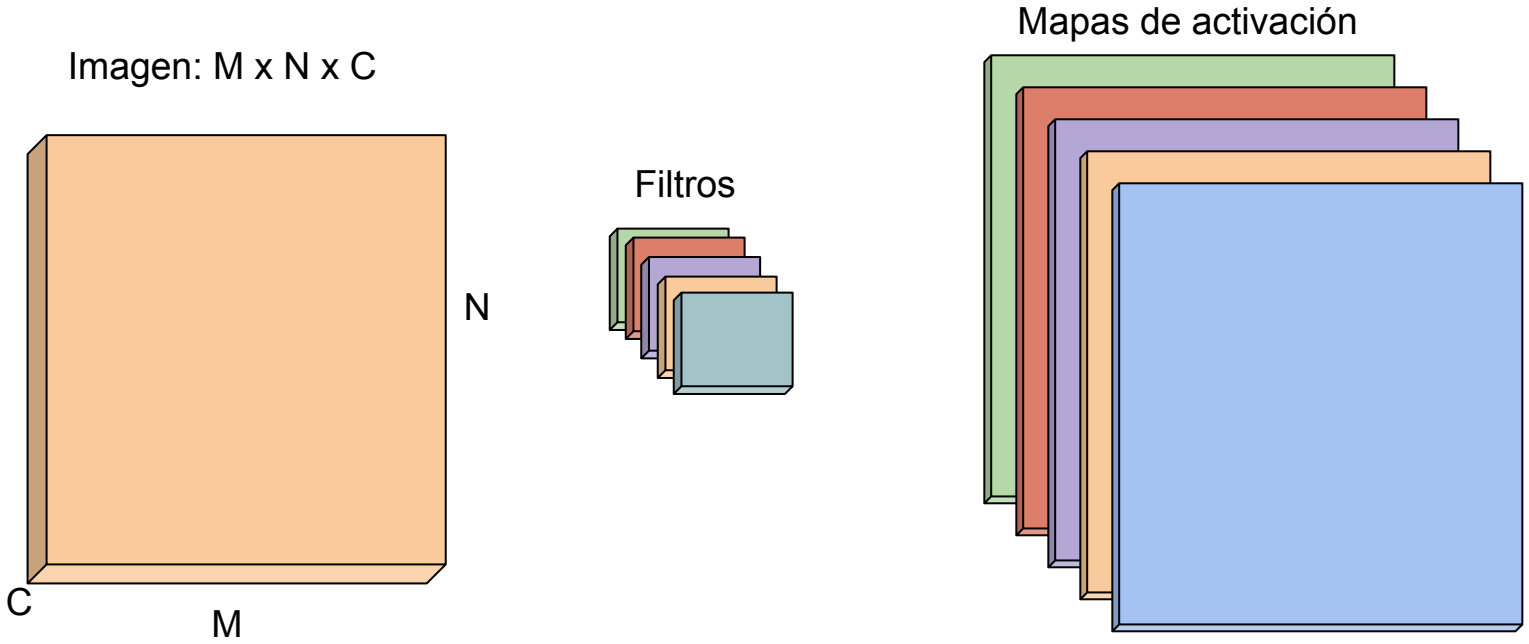
Capa de convolución



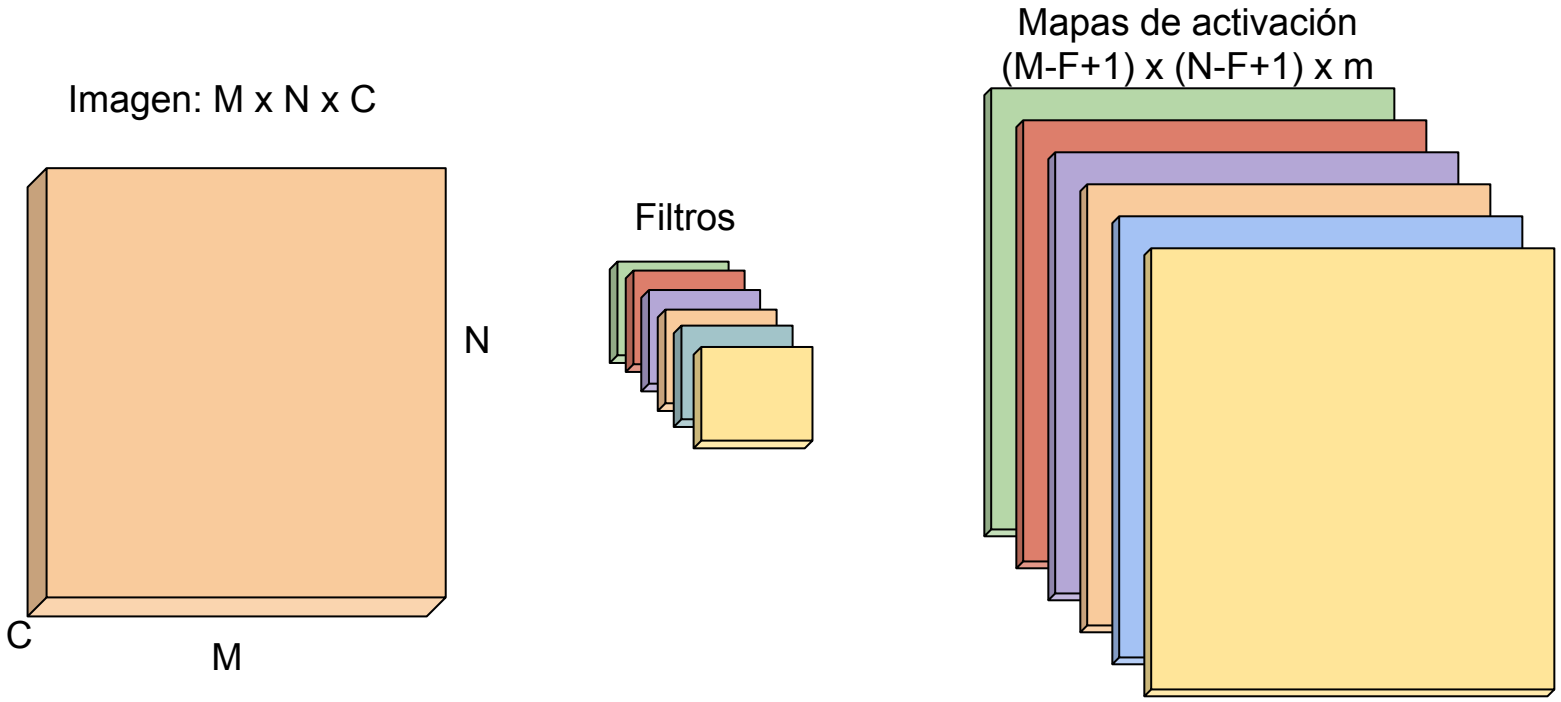
Capa de convolución



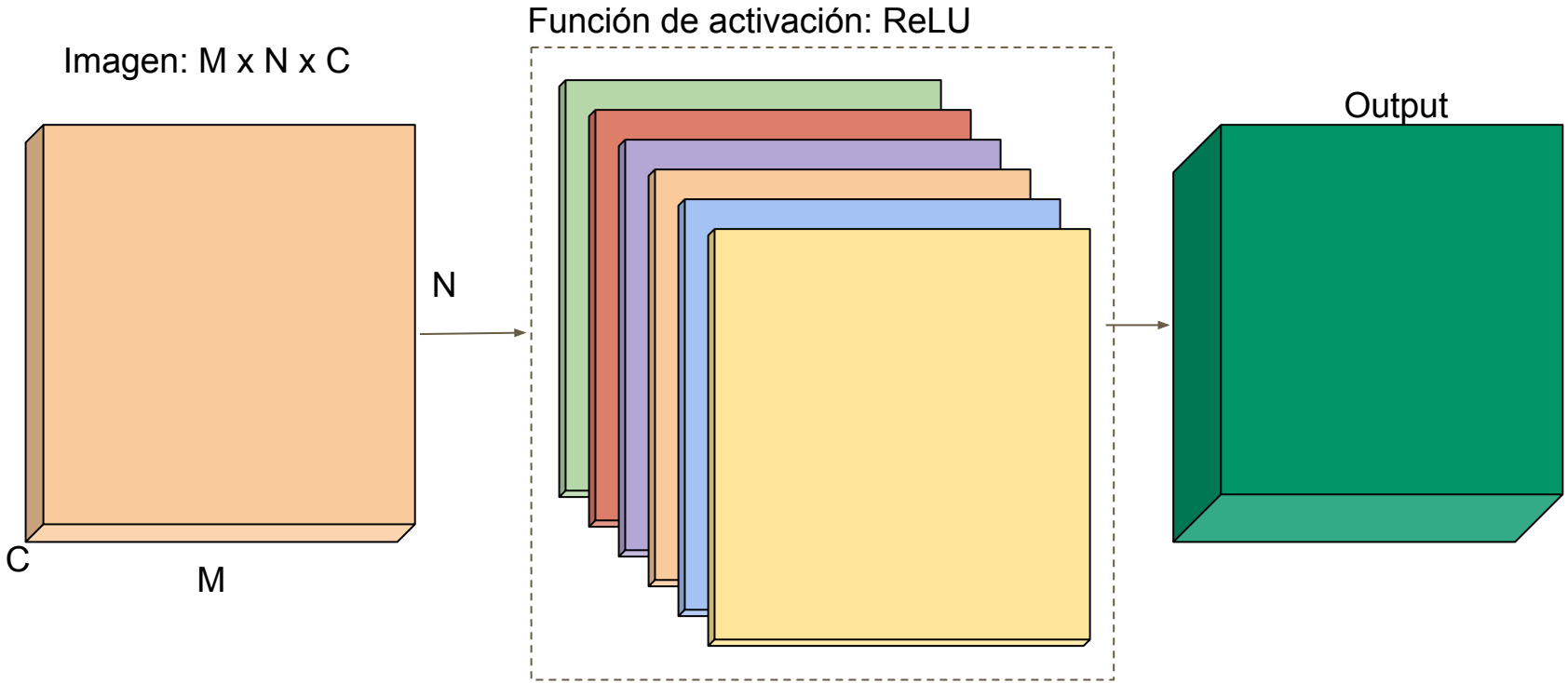
Capa de convolución



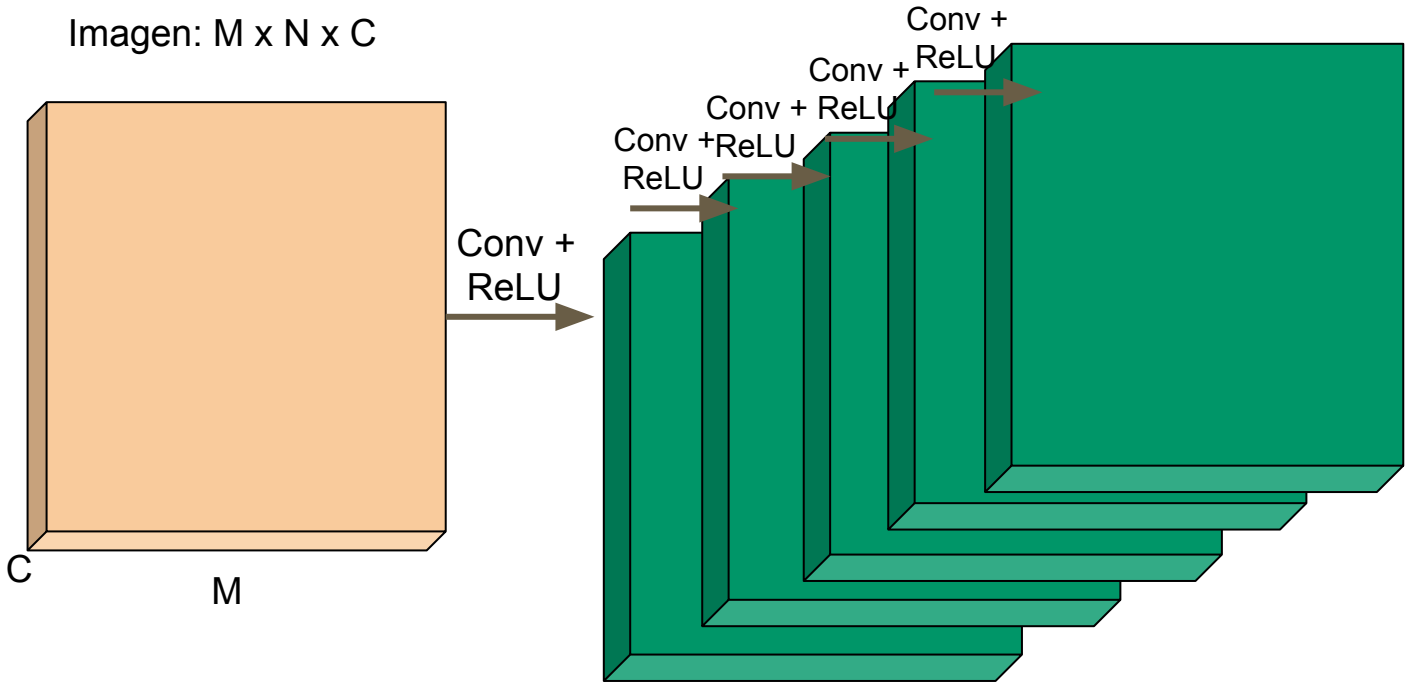
Capa de convolución



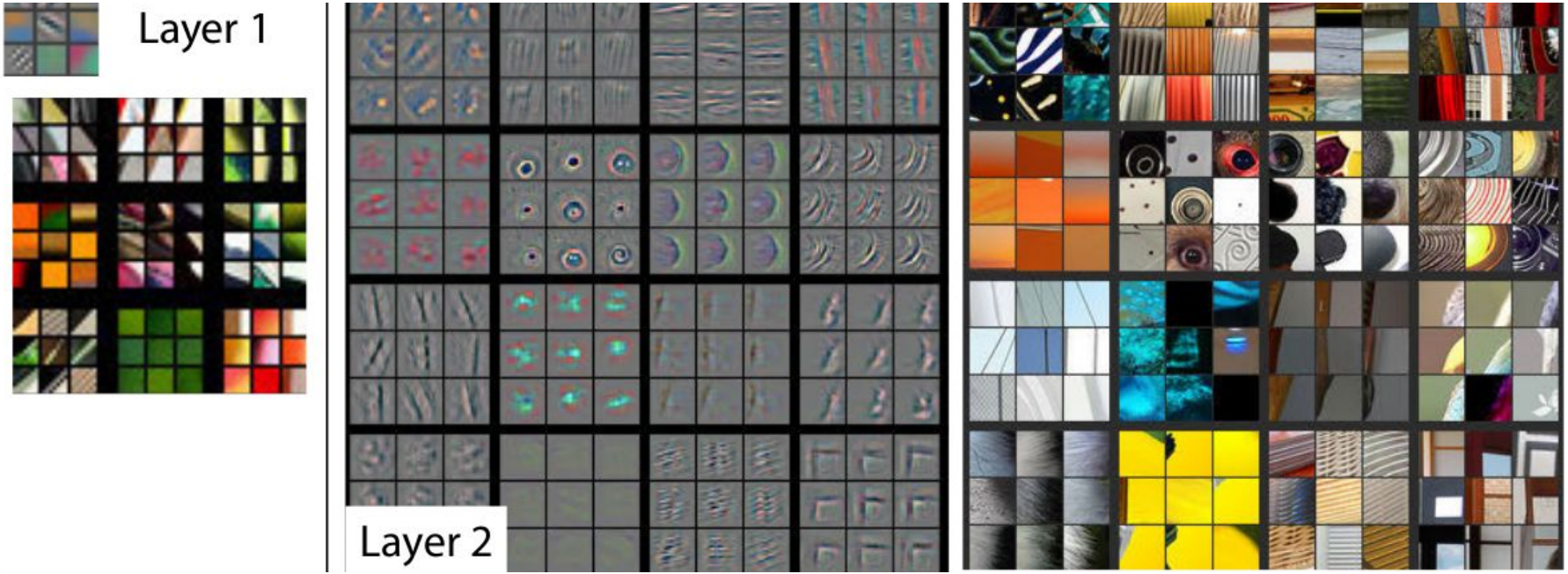
Capa de convolución: Convolución + Activación



Capa de convolución: Convolución + Activación

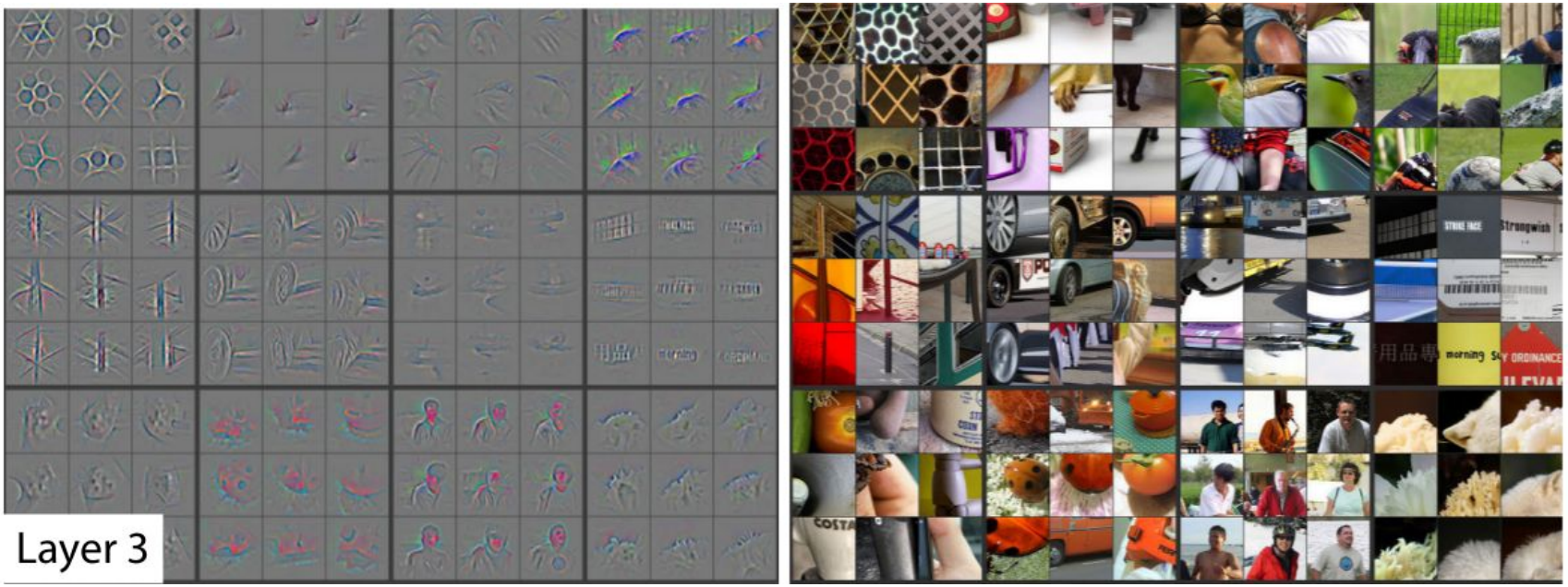


Visualización



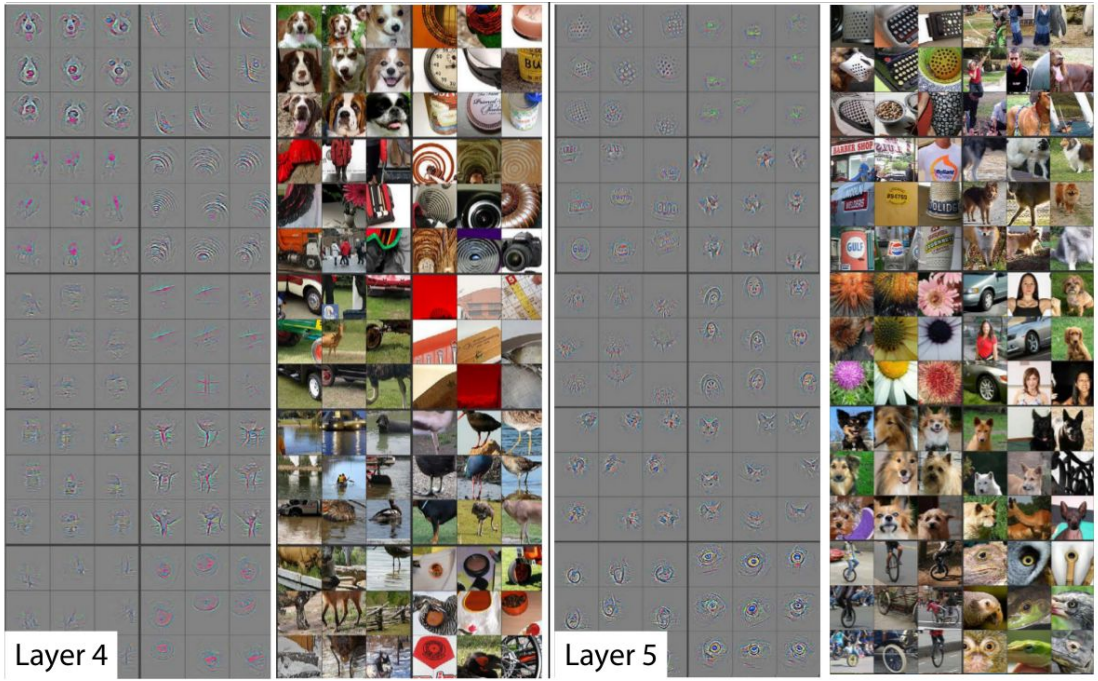
Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Visualización



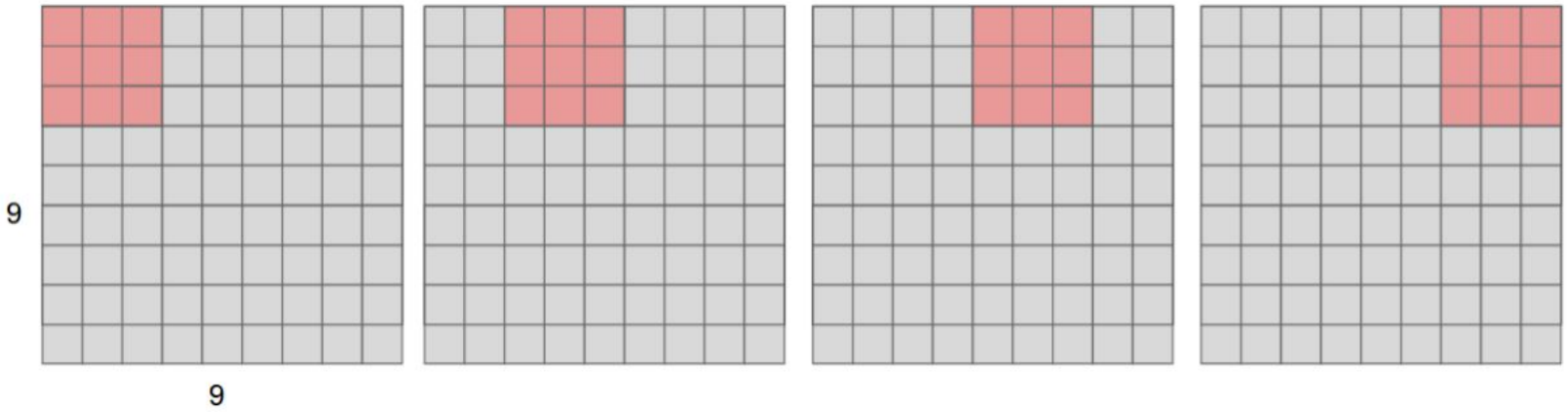
Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Visualización



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Dimensiones

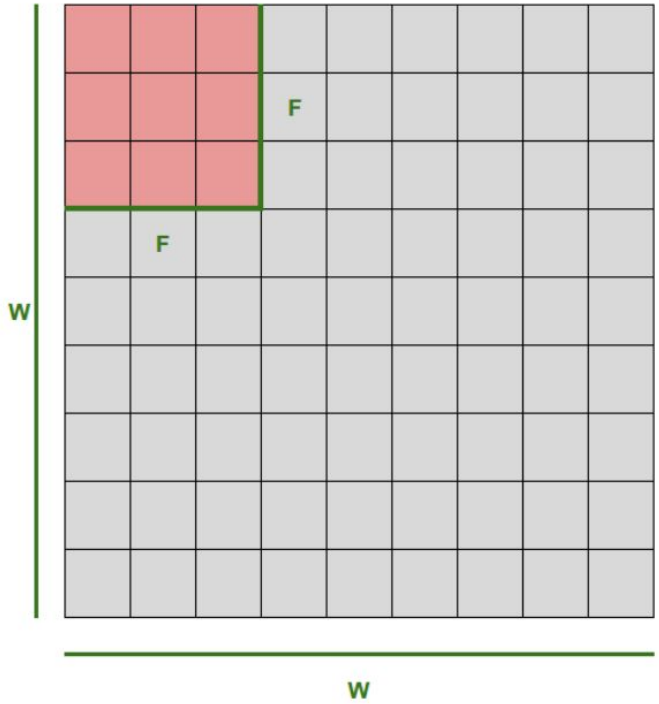


(a) Imagen de entrada: 9×9,
Filtro: 3×3 Stride: 2,
Salida: 4×4.

Dimensiones

- Tamaño de salida:
  - (W-F) / stride + 1
  - Ej: (9-3) / 2 + 1 = 4

- Stride = 4 ??



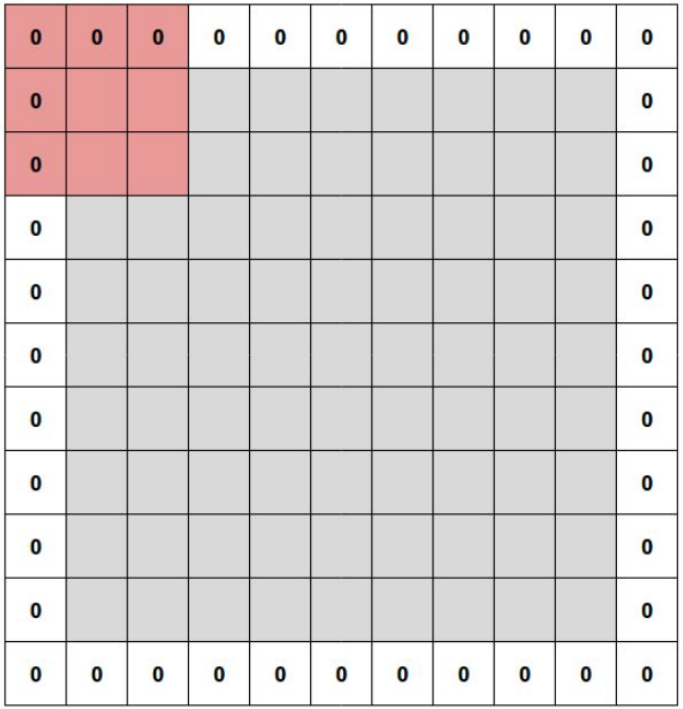
(b) Tamaño de salida: $(W-F)/\text{stride}+1$,
Ej: $(9-3)/2+1=4$

Zero-padding

- Imagen de entrada: 9 × 9
- Filtro: 3 × 3
- Stride: 1
- Pad de borde 1
- Salida: 9 × 9

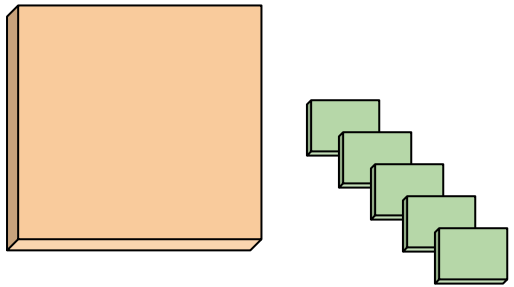
Normalmente se utilizan capas de convolución con

- stride 1,
- filtros de tamaño F × F
- zero-padding de (F − 1)/2 (same)



Ejemplo

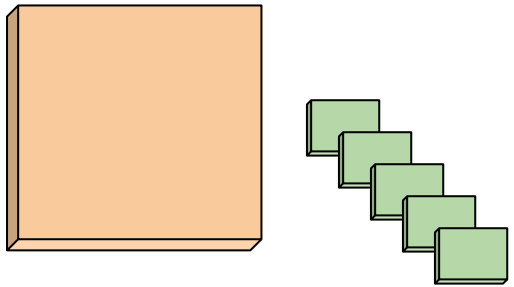
- Tamaño de entrada: 32x32x3
- Filtros:
  - Cantidad: 10
  - Tamaño: 5x5
  - Stride: 1
  - Pad: 2



- Tamaño de salida?

Ejemplo

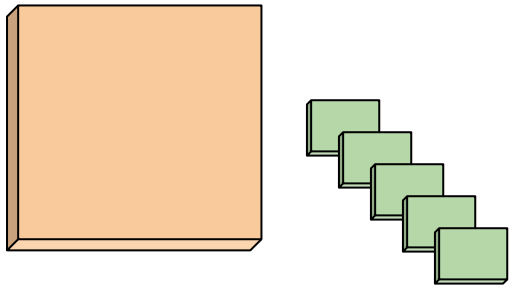
- Tamaño de entrada: 32 x 32 x 3
- Filtros:
  - Cantidad: **10**
  - Tamaño: **5**x**5**
  - Stride: **1**
  - Pad: **2**



- Tamaño de salida:
  - **(32 + 2*2 - 5)/1+1=32** espacial => 32 x 32 x **10**

Ejemplo

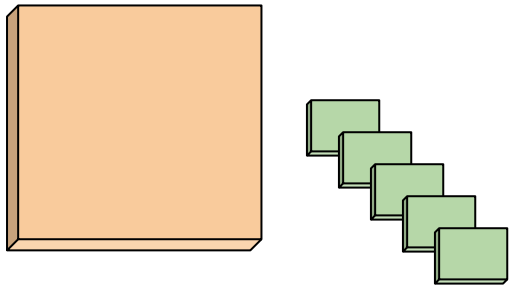
- Tamaño de entrada: 32x32x3
- Filtros:
  - Cantidad: 10
  - Tamaño: 5x5
  - Stride: 1
  - Pad: 2



- Número de parámetros en esta capa?

Ejemplo

- Tamaño de entrada: 32 x 32 x 3
- Filtros:
  - Cantidad: **10**
  - Tamaño: **5**x**5**
  - Stride: **1**
  - Pad: **2**



- Número de parámetros en esta capa?
  - Cada filtro tiene: **5x5x3+1**(bias) **= 76** parámetros => 76 x **10** = 760

Pooling layer

- Comprime (sub-muestreo) de la representación
- Opera en cada mapa de activación (canal) por separado
- Su objetivo es reducir la cantidad de parámetros de la CNN - ayudando a prevenir el overfitting.

Pooling layer

- Max-pooling es el más utilizado.
- Usualmente se utiliza max pooling de 2x2, con stride de 2
- ¿Parámetros?

CNNs

- Las arquitecturas de las CNNs son una lista de bloques de capas que transforman la imagen de entrada en sucesivos volúmenes de salida.
- Los bloques de las CNNs están compuestos típicamente por un conjunto de capas comunes: Conv, ReLU, POOL, FC
- Cada capa puede tener parámetros (Conv, FC) o no (Pool, ReLU)
- Cada capa puede o no tener hiperparametros adicionales (e.g. CONV/FC/POOL tienen, RELU no)

Quiz time!

1. ¿Verdadero o Falso?
    - Las redes convolucionales fueron diseñadas para capturar las relaciones entre secciones distantes de las imágenes que procesan.
    - En una CNN, agregar capas de max-pooling ayuda a que el modelo generalice mejor.
2. ¿Cuáles son las dimensiones del volumen resultante de aplicar a una imagen RGB de 16*16 píxeles, una capa de convolución con 10 filtros de dimensiones (5, 5), con stride de tamaño 1 y zero padding de tamaño 2?
3. ¿Y si luego aplicamos max pooling de 2*2?
4. ¿Cuántos parámetros tiene un bloque convolucional que recibe una entrada de dimensiones (512,512,3) con 10 filtros (3,3), padding y stride de tamaño 1 y max-pooling de (2,2)?

Ejemplo

- Clasificación CIFAR-10
- Arquitectura: [conv-relu-conv-relu-pool]x3-fc-softmax
  - 17 capas
  - 7000 parameters
  - 3x3 convolutions
  - 2x2 pooling

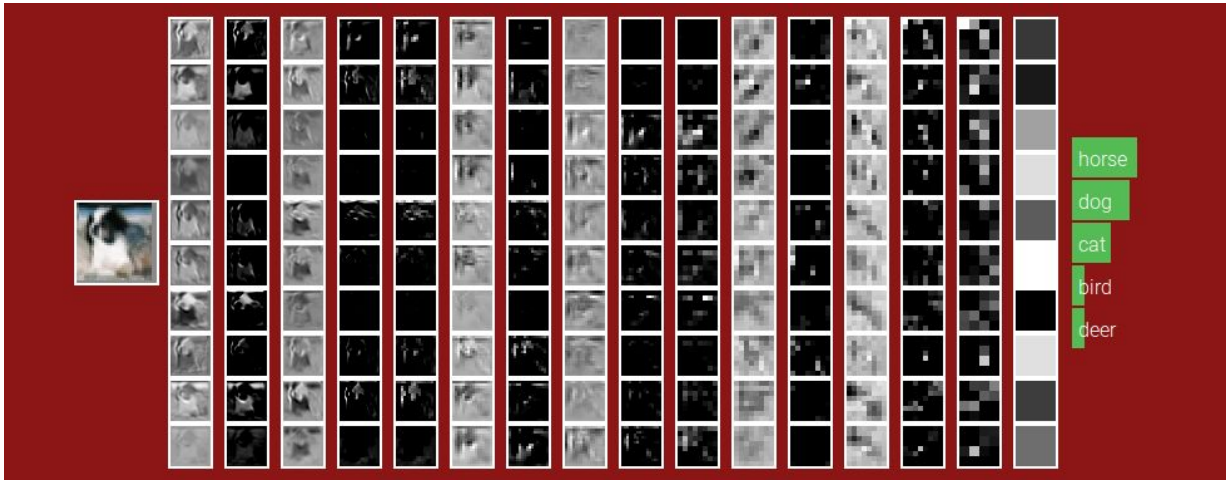
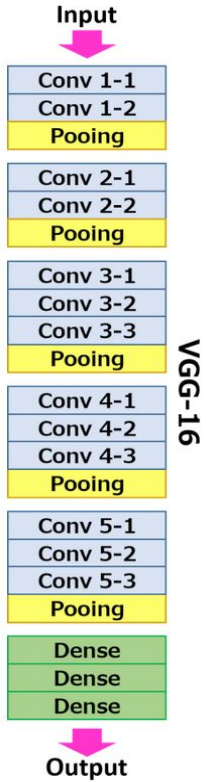


Imagen tomada de http://cs231n.stanford.edu/

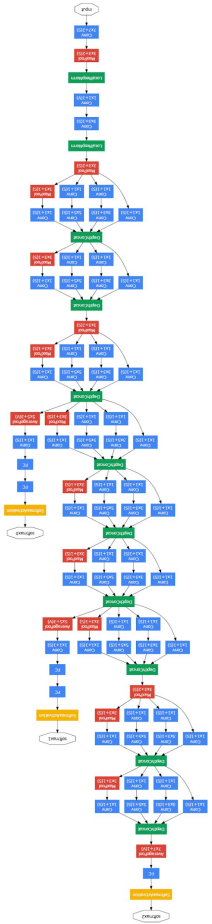
Arquitecturas - VGG

- “Very Deep Convolutional Networks for Large-Scale Visual Recognition”, Karen Simonyan y Andrew Zisserman, 2014.
- Primer y segundo puesto del challenge ImageNet ILSVRC-2014 - clasificación y localización.
- Arquitectura muy homogénea.
- Demuestran que la profundidad es un factor importante para lograr buena performance (utilizan hasta 19 capas)



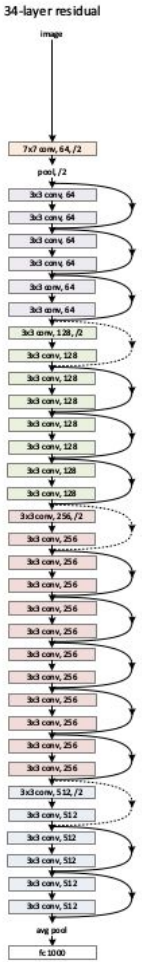
Arquitecturas - GoogLeNET

- “Going Deeper with Convolutions”, Christian Szegedy et. al, 2014.
- Ganadores del challenge ImageNet ILSVRC-2014 - detección.
- Introducen el bloque de “inception”, que que permite aplicar diferentes tamaños de filtros en la misma capa de convolución - dejando esta “decisión” al proceso de entrenamiento.
- Utilizan clasificadores intermedios que ayudan a prevenir el problema de vanishing gradient.



Arquitecturas - Resnet

- Deep Residual Learning for Image Recognition, Kaiming He et al., 2015.
- Ganadores de ILSVRC 2015: detección, y detección con localización.
- Investigan el problema de la degradación en performance al hacer las CNNs más profundas.
- Intuición: una red poco profunda es una subred de la red completa. Sin embargo, con demasiada profundidad, la performance se degrada.
- Proponen agregar conexiones residuales, que facilitan que la red aprenda la identidad cuando sea necesario.
- Permite redes más profundas sin el problema del vanishing gradient.



Transfer Learning y Fine-Tuning

Algunas consideraciones prácticas:

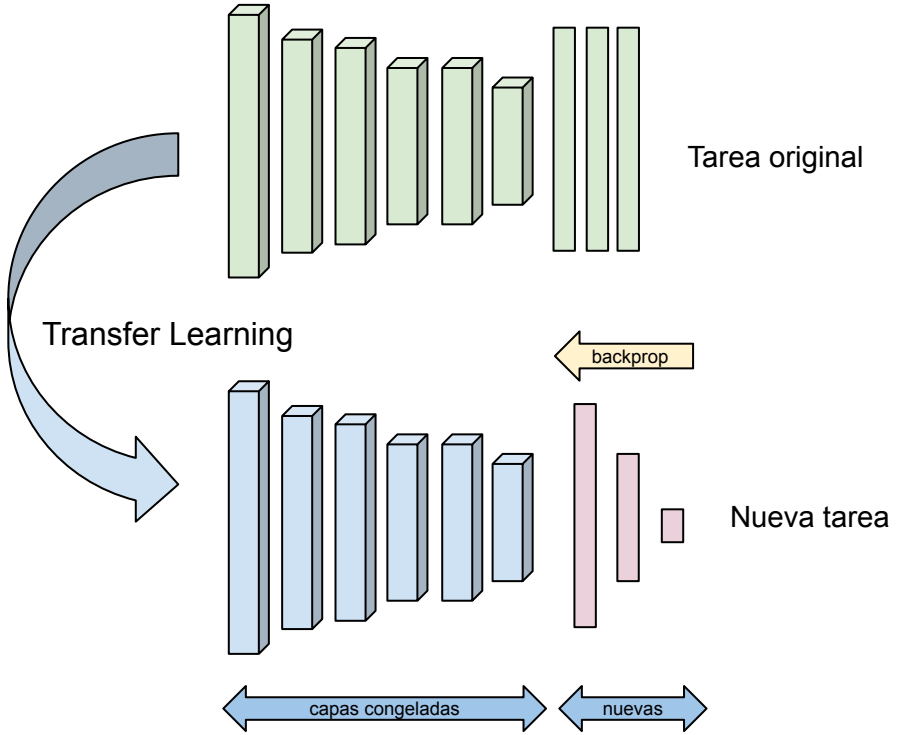
- Estas arquitecturas requieren de muchísimos ejemplos de entrenamiento. Si bien, las imágenes abundan, el proceso de etiquetado suele ser costoso.
- Se requiere hardware especial para acelerar los procesos (GPU).
- Desarrollo, búsqueda de hiperparámetros y evaluación terminan siendo restrictivos.

Transfer Learning y Fine-Tuning

- Las técnicas de ***transfer learning*** y ***fine tuning*** proponen reutilizar redes pre-entrenadas.
- Se sustituyen las capas finales por nuevas capas que se ajustarán a nuevos problemas.
- Las capas anteriores (con sus pesos ya pre-entrenados) se reutilizan:
  - transfer learning: las “congela”
  - fine tuning: las sigue entrenando

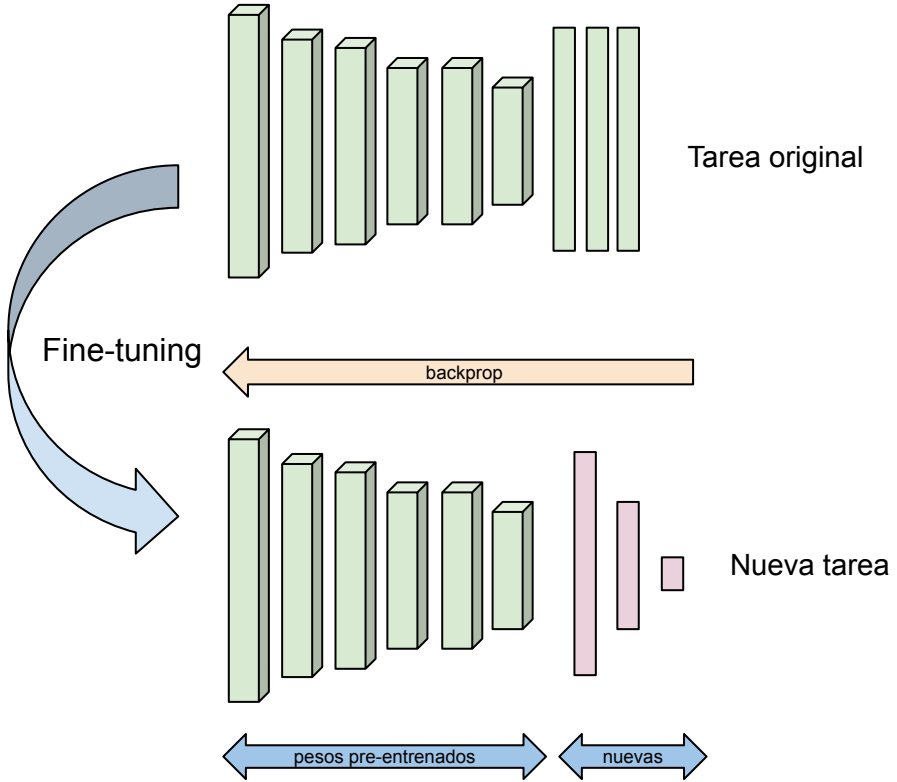
Transfer Learning y Fine-Tuning

- Transfer learning:
  - las capas anteriores se “congelan”, dejando fijos los pesos. Sólo se entrenan las nuevas capas finales.
  - es más rápido que el fine-tuning, porque hay menos parámetros para ajustar.



Transfer Learning y Fine-Tuning

- Fine tuning:
  - no se “congela” ninguna capa.
  - Se sustituyen las capas finales por nuevas capas, y se entrenan todos los pesos.
  - Es más costoso que el transfer learning, pero puede dar mejores resultados, ya que podemos modificar los pesos de etapas tempranas para que se ajusten mejor al nuevo problema.



Bibliografía/recursos

[1] CS231n Convolutional Neural Networks for Visual Recognition - Stanford CS class

[2] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[3] Chollet, Francois. Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG, 2018.

[4] Paul Viola , Michael Jones. Rapid object detection using a boosted cascade of simple features (2001).