

THE BRIDGE

Deep Learning

Deep Learning

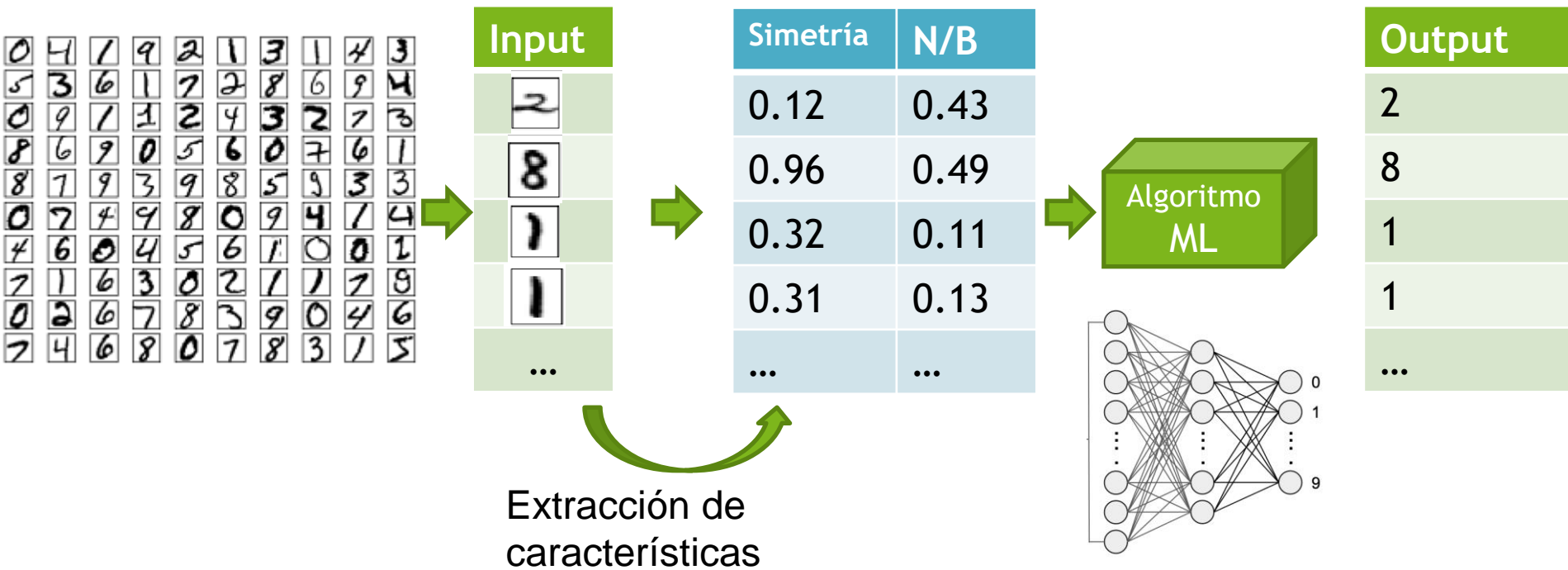
- Machine Learning vs. Deep Learning



- En Machine Learning, se seleccionan manualmente las características
- Con el aprendizaje profundo, los pasos de extracción de características y modelización son automáticos (redes neuronales **profundas**)
- Para utilizar deep learning, es necesario disponer de GPUs y una gran cantidad de datos etiquetados

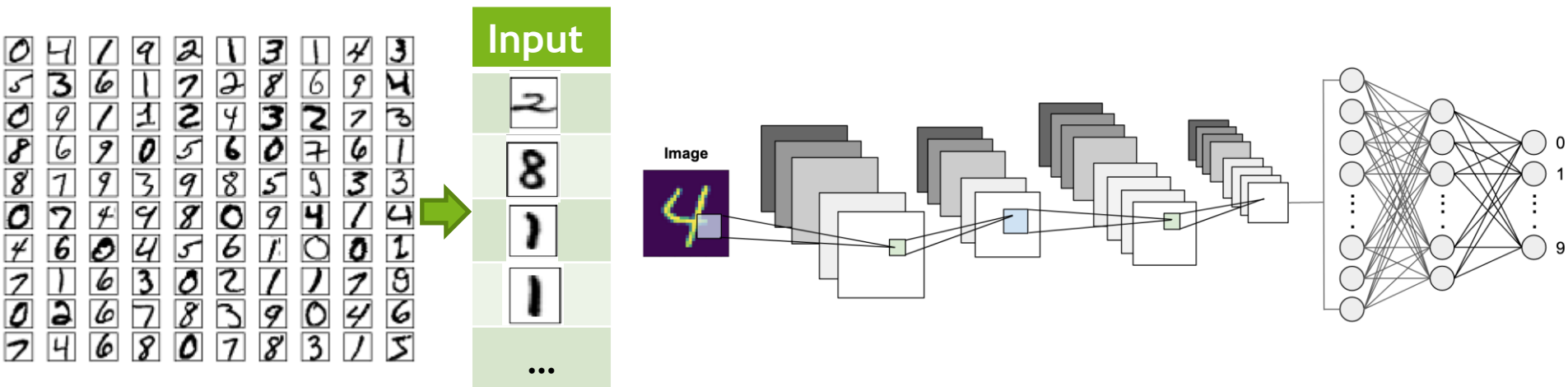
Deep Learning

- Machine Learning vs. Deep Learning



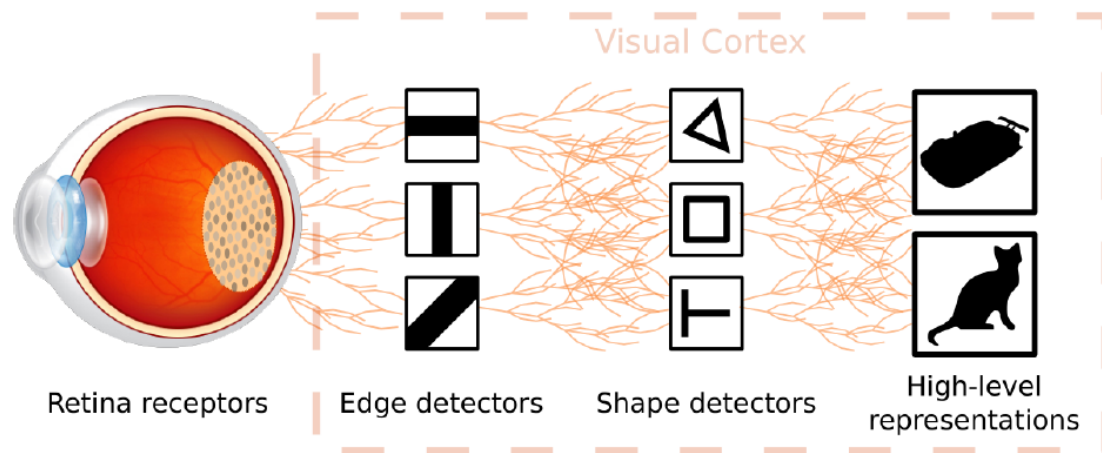
Deep Learning

- Machine Learning vs. Deep Learning



Redes Neuronales Convolucionales (CNN)

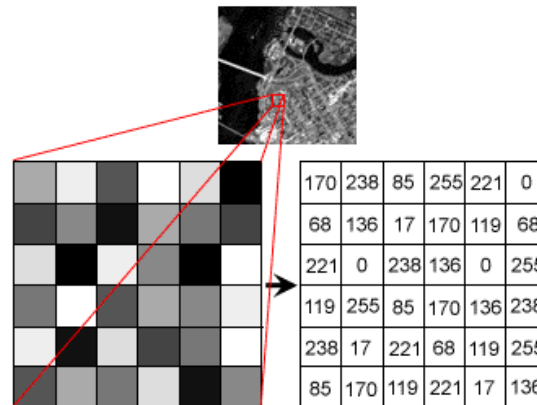
- Uno de los algoritmos más populares de redes neuronales profundas
- Son especialmente eficaces en la **clasificación de imágenes**, aunque también pueden utilizarse con series temporales o señales de audio y vídeo
- Imitan el funcionamiento del sistema visual humano



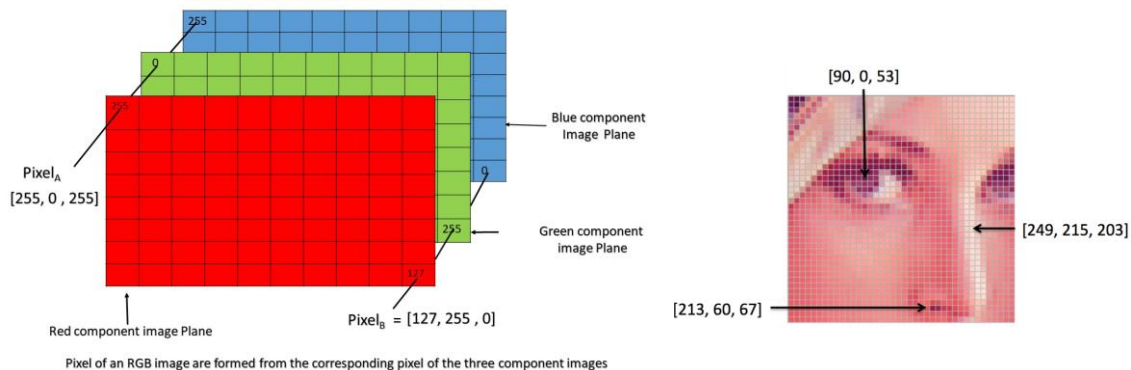
- Son robustas ante cambios de tamaño, contraste, rotación u orientación

Redes Neuronales Convolucionales (CNN)

- Representación digital de imágenes: matriz de píxeles con valores de intensidad entre 0 y 255



- Las imágenes en color se representan con tres capas superpuestas (RGB)

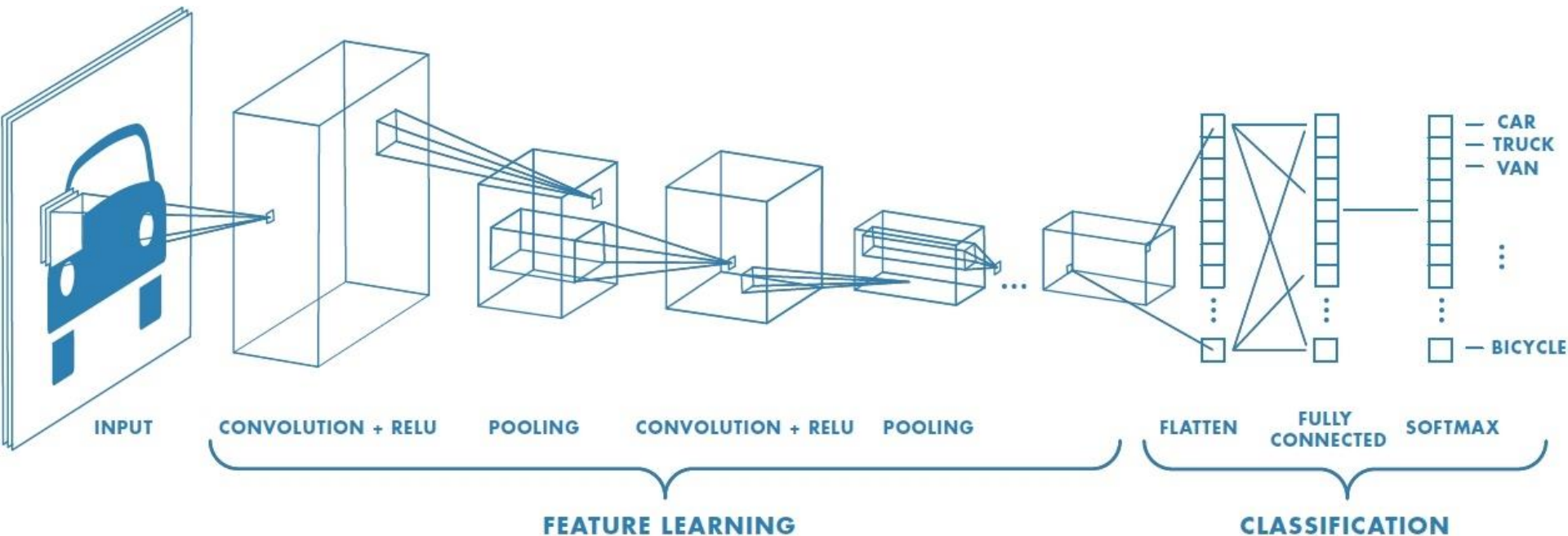


Redes Neuronales Convolucionales (CNN)

Etapas en CNN:

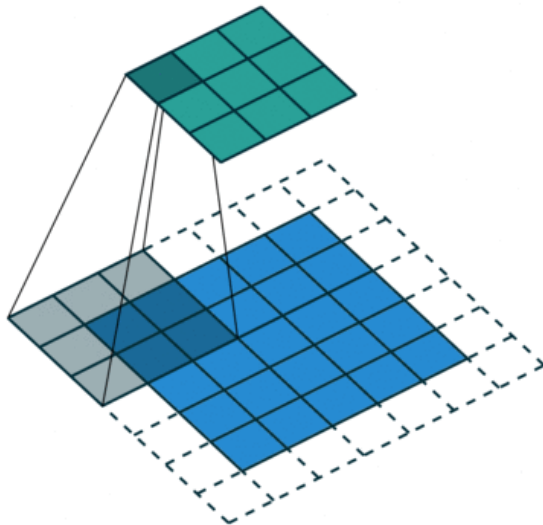
- Convolución
- Normalización (ReLU)
- Pooling
- Regularización

Pensemos en
capas, no en
neuronas



Redes Neuronales Convolucionales (CNN)

- Las CNN realizan operaciones de convolución, una técnica ampliamente utilizada en procesamiento de señales e imágenes
- La convolución permite preservar la relación entre diferentes partes de una imagen
- Para realizar las convoluciones, se utilizan pequeñas matrices denominadas kernels, que recorren la imagen original, aplican productos escalares y producen una nueva imagen con características diferentes



7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

1	0	-1
1	0	-1
1	0	-1

=

6		

$$\begin{aligned} &7 \times 1 + 4 \times 1 + 3 \times 1 + \\ &2 \times 0 + 5 \times 0 + 3 \times 0 + \\ &3 \times -1 + 3 \times -1 + 2 \times -1 \\ &= 6 \end{aligned}$$

Redes Neuronales Convolucionales (CNN)

- Dependiendo del kernel, se resaltarán diferentes aspectos de una imagen



*

-1	-2	-1
0	0	0
1	2	1

(bordes)

=



*

-2	-1	0
-1	1	1
0	1	2


(relieve)


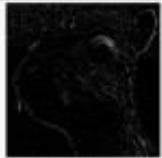

=



Redes Neuronales Convolucionales (CNN)

- En una capa convolucional, suelen utilizarse varios kernels
- Los pesos del kernel se aprenden durante el entrenamiento (backpropagation)

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	

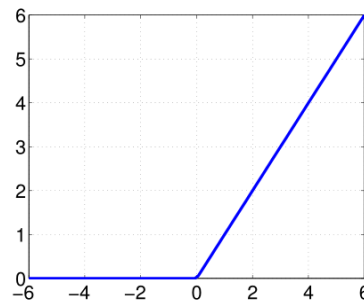
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Redes Neuronales Convolucionales (CNN)

- La función de activación más utilizada tras aplicar la convolución se denomina Rectifier Linear Unit (ReLU), que facilita el entrenamiento

$$f(x) = \max \{0, x\}$$

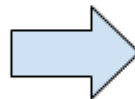


IMAGEN

		0,6	0,6		
	0,6			0,6	
	0,6	0,6	0,6	0,6	
	0,6			0,6	

KERNEL

1	0	-1
2	0	-2
1	0	-1



CONVOLUCION
DEL KERNEL

-1,2	-0,6	0,6	1,2
-1,2	0,6	-0,6	1,2
-1,2	1,2	-1,2	1,2
-0,6	1,2	-1,2	0,6



APLICO RELU

0	0	0,6	1,2
0	0,6	0	1,2
0	1,2	0	1,2
0	1,2	0	0,6

Redes Neuronales Convolucionales (CNN)

- ¿Cuál es el resultado de la siguiente convolución?

3	1	1	2	8	4
1	0	7	3	2	6
2	3	5	1	1	3
1	4	1	2	6	5
3	2	1	3	7	2
9	2	6	2	5	1

Original image 6x6

“Convolution”



1	0	-1
1	0	-1
1	0	-1

Filter 3x3

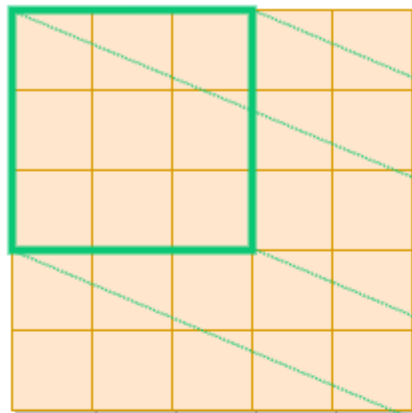
=

-7	...		
...	...		

Output 4x4

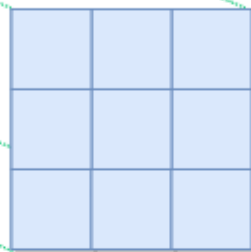
Result of the element-wise product and sum of the filter matrix and the original image

Input D x D: 5 x 5

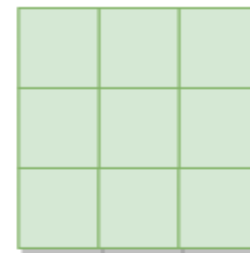


Padding VALID
Output dimension = $D - N + 1$
 $5 - 3 + 1 = 3$

Filter N x N: 3 x 3

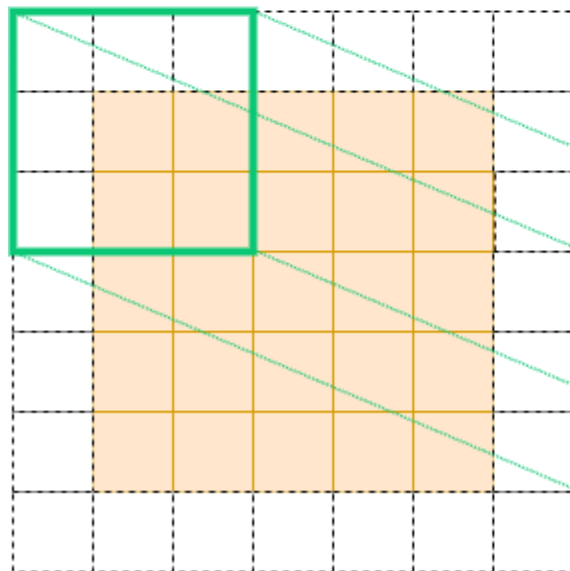


Output: 3 x 3



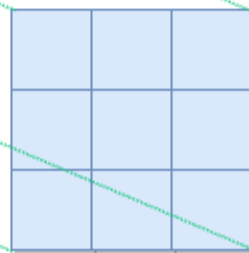
Podemos elegir diferentes estrategias de convolución

Input D x D: 5 x 5
Plus added padding of size 1

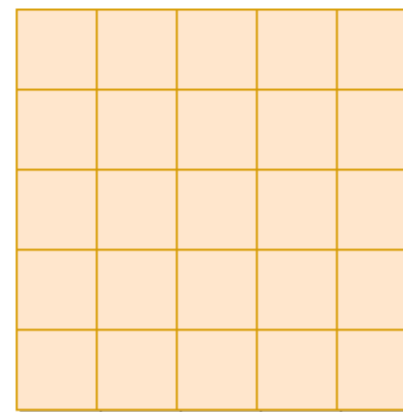


Padding SAME
Output dimension = Input dimension

Filter N x N: 3 x 3



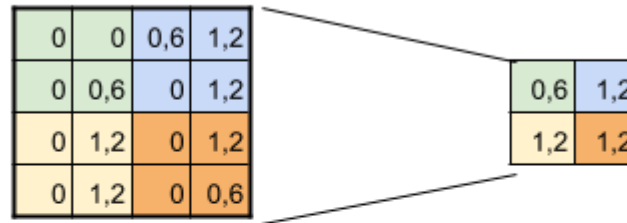
Output: 5 x 5



Redes Neuronales Convolucionales (CNN)

Pooling

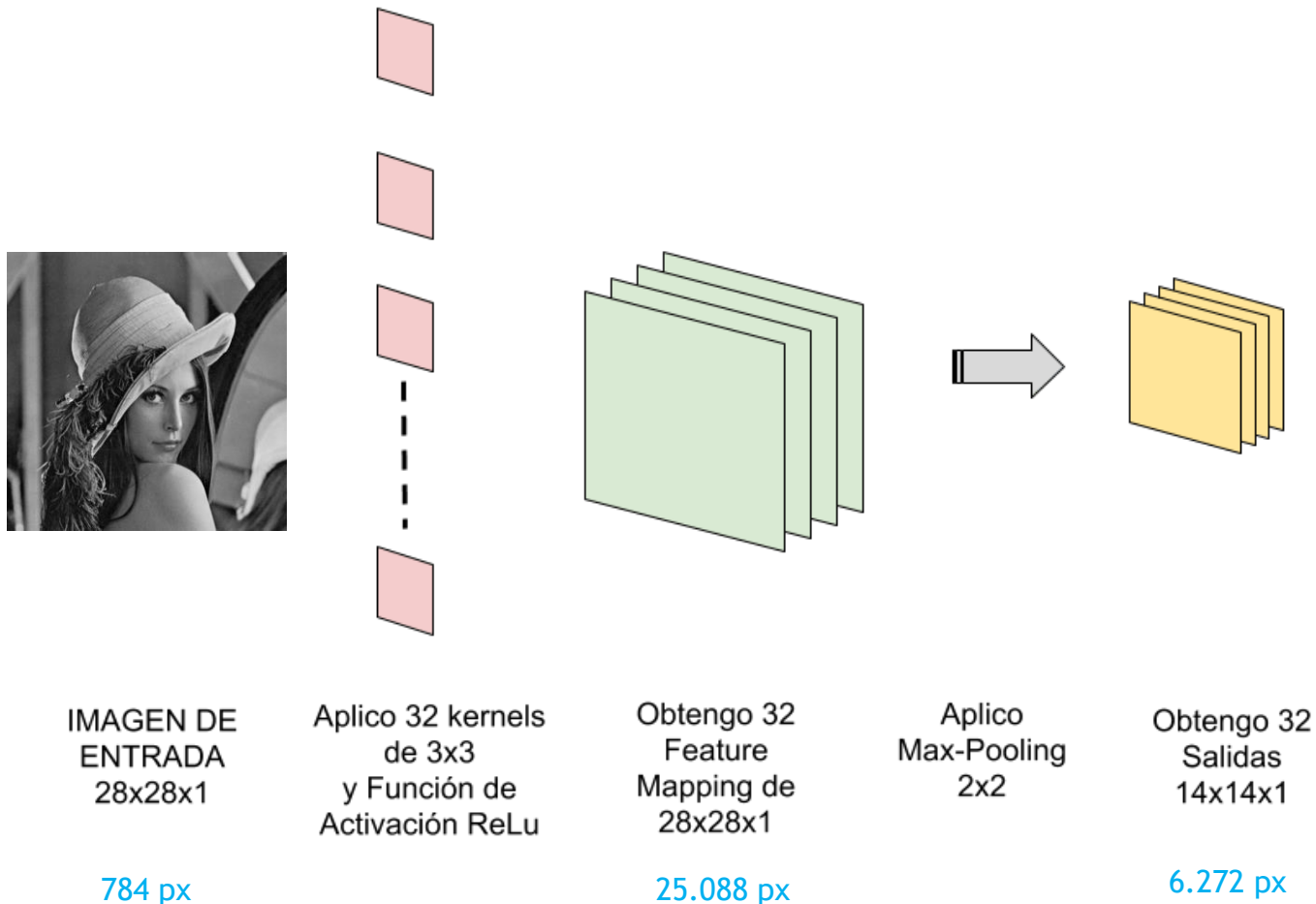
- Calcula estadísticas por grupos de píxeles
- Reduce complejidad computacional y evita el overfitting
- Es invariante al escalado y pequeñas traslaciones, manteniendo las características más importantes que detectó cada convolución
- Suele utilizarse el max-pooling, que mantiene las características más destacadas



SUBSAMPLING:
Aplico Max-Pooling de 2x2
y reduzco mi salida a la mitad

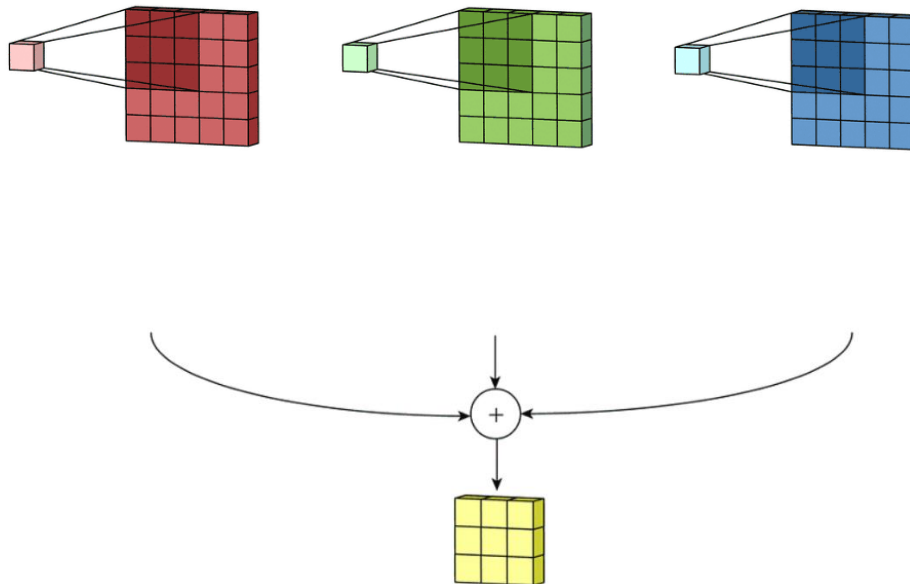
Redes Neuronales Convolucionales (CNN)

Ejemplo



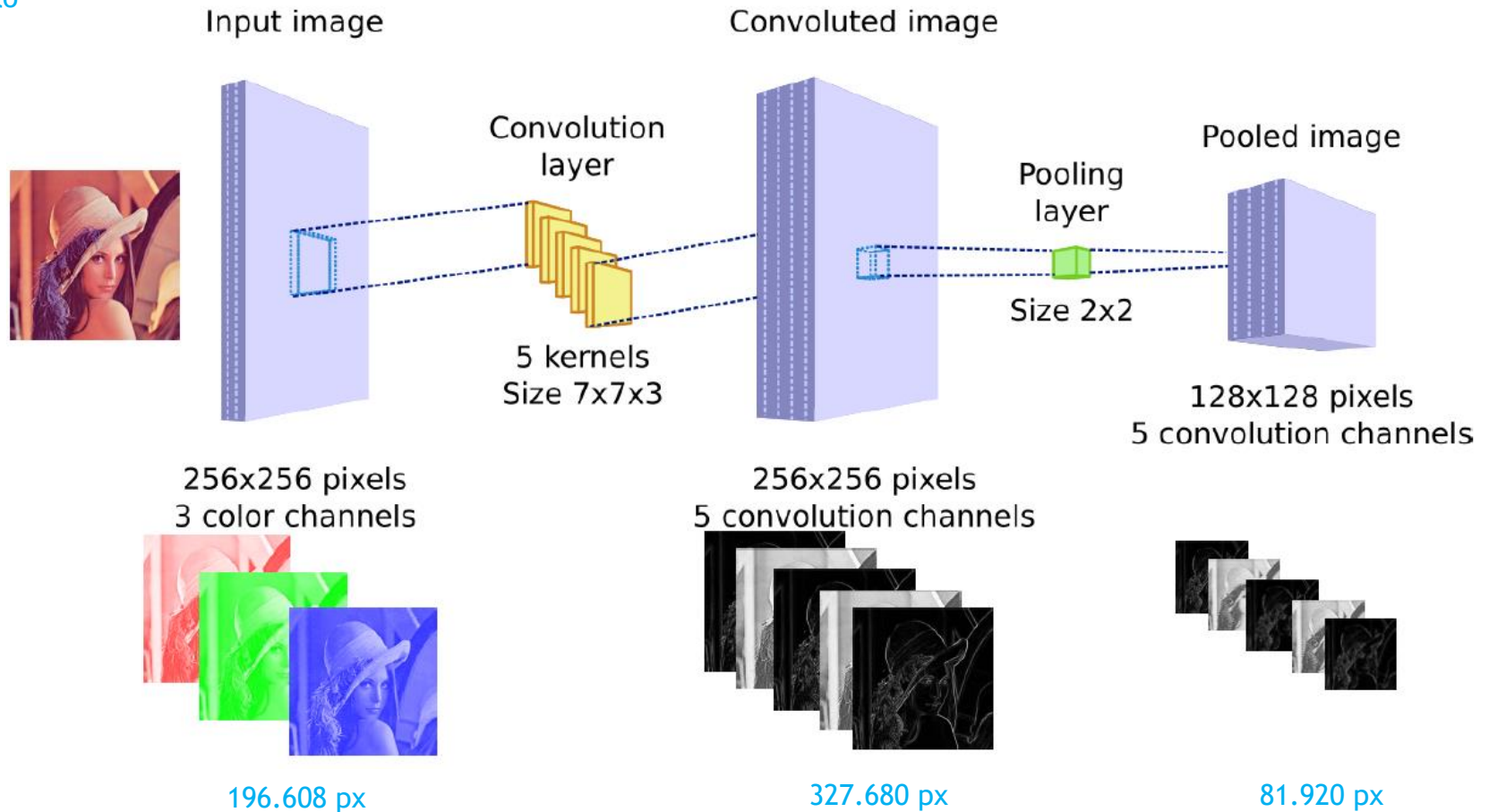
Redes Neuronales Convolucionales (CNN)

- Con imágenes RGB, cada kernel se aplica a cada canal separadamente, y posteriormente se suman los resultados



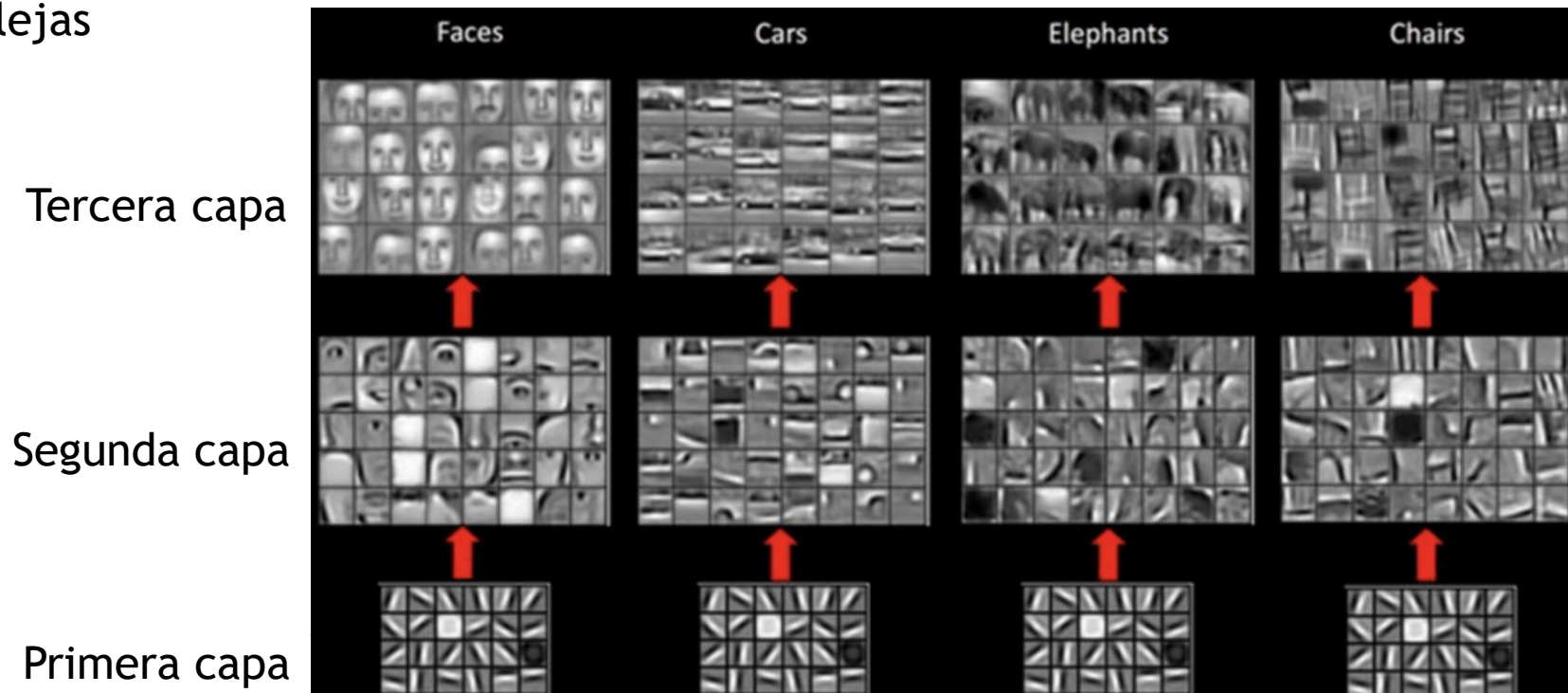
Redes Neuronales Convolucionales (CNN)

Ejemplo



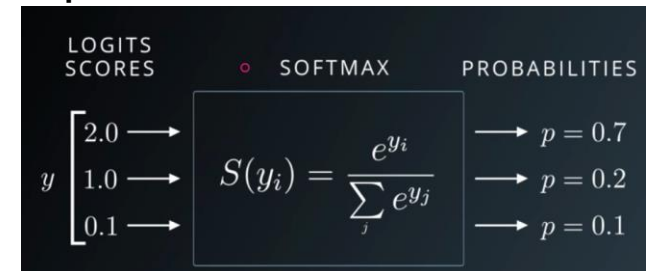
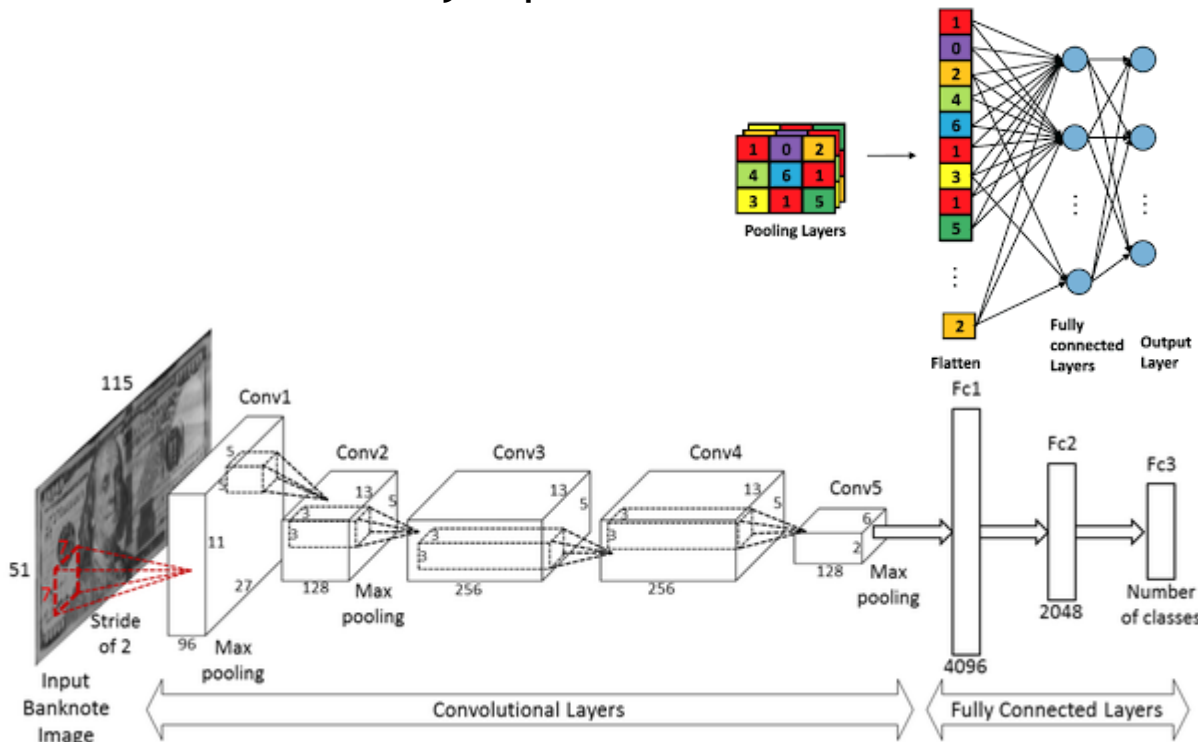
Redes Neuronales Convolucionales (CNN)

- La primera convolución es capaz de detectar características primitivas como líneas o curvas
- A medida que se realicen más convoluciones, la red reconocerá formas más complejas



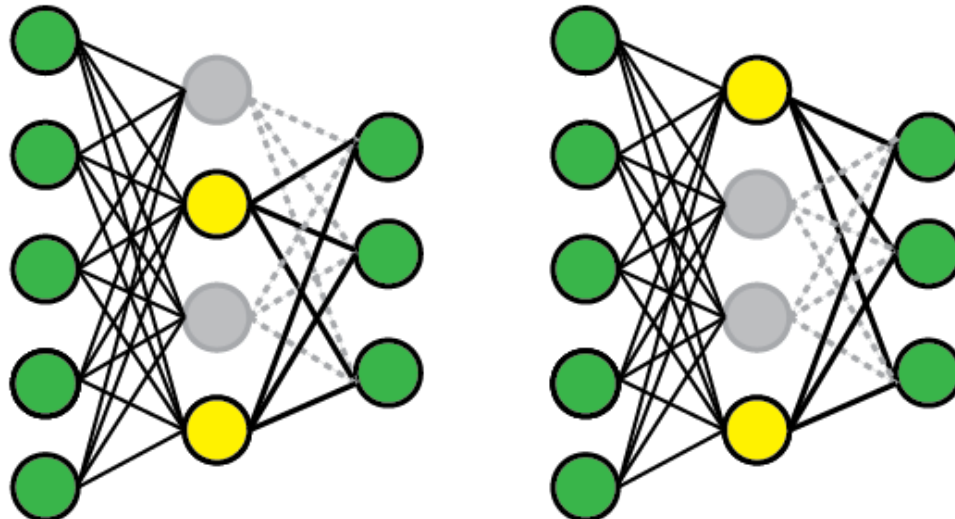
Redes Neuronales Convolucionales (CNN)

- Tras las capas de convolución+pooling, se aplanan los datos y se utiliza una red neuronal “tradicional”, teniendo una neurona de salida por cada categoría
- Al final de la red, se aplica una función denominada *softmax*, encargada de transformar los valores de entrada a probabilidades
- La clase con mayor probabilidad será el resultado de la predicción



Redes Neuronales Convolucionales (CNN)

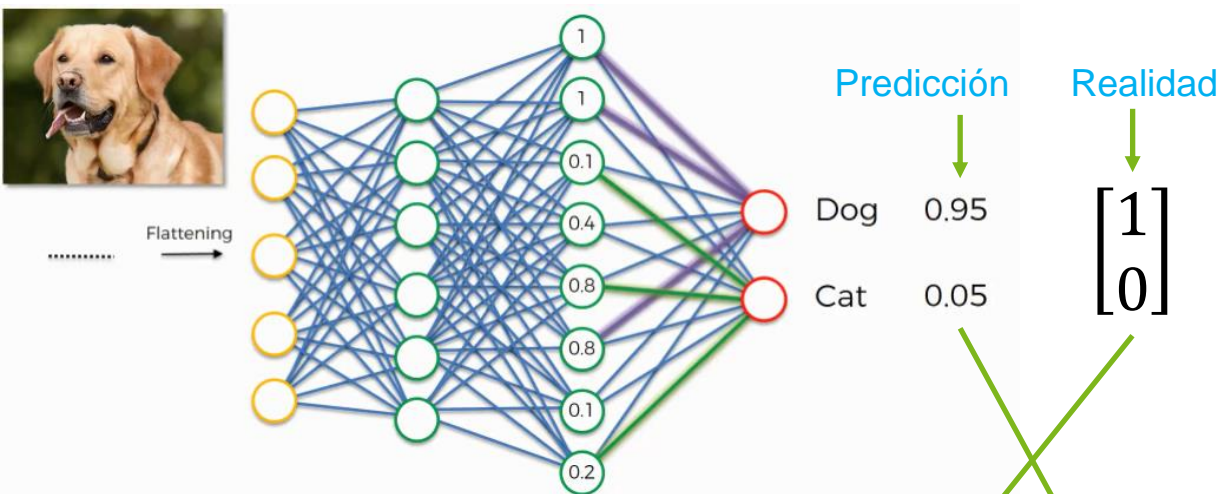
- En la red neuronal final, se suele utilizar un proceso de regularización conocido como **dropout**, con el objetivo de evitar overfitting
- Esta técnica deshabilita aleatoriamente neuronas durante el entrenamiento, forzando a la red a aprender múltiples representaciones independientes del mismo dato



Redes Neuronales Convolucionales (CNN)

Función de error: Cross Entropy

- Suele emplearse esta función de error para entrenar la red neuronal

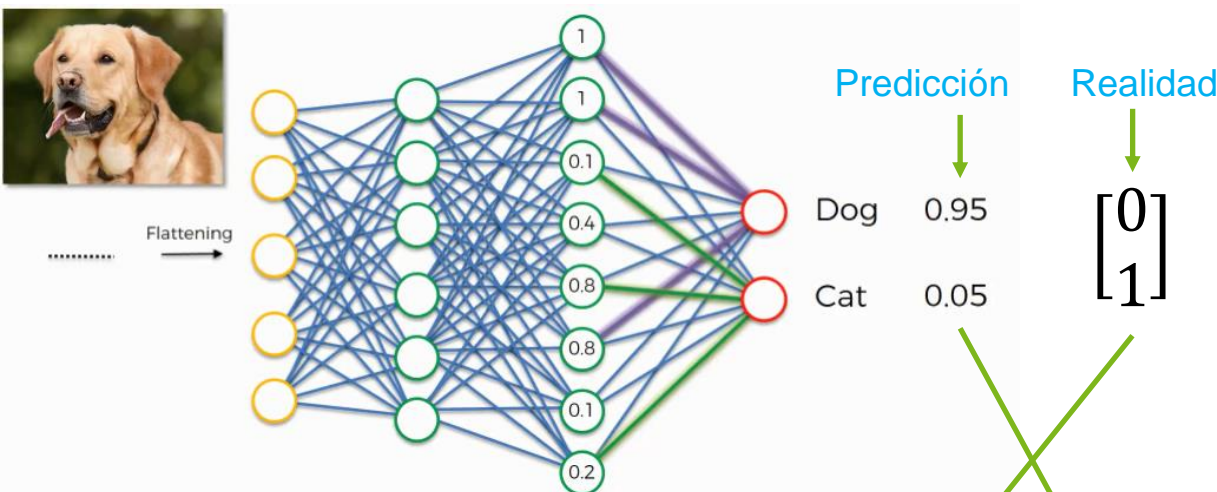


$$D(S, L) = - \sum L_i \log(S_i) = - 1 \cdot \log(0.95) = 0.05$$

Redes Neuronales Convolucionales (CNN)

Función de error: Cross Entropy

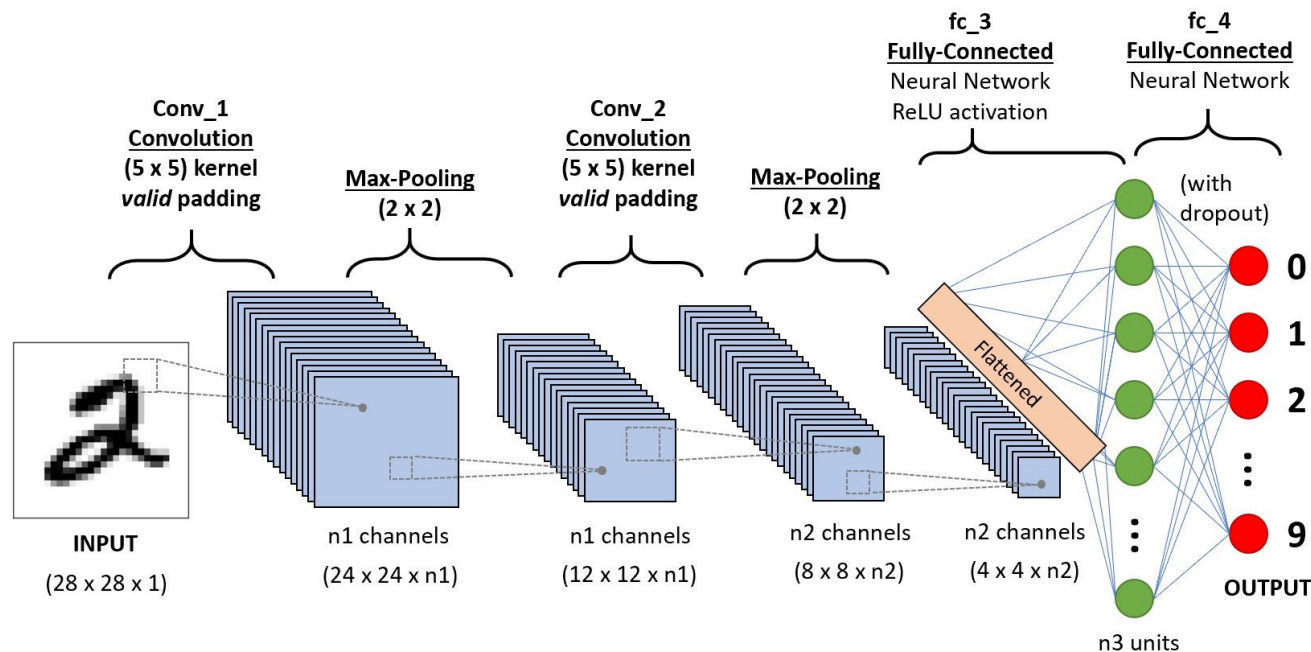
- Suele emplearse esta función de error para entrenar la red neuronal



$$D(S, L) = - \sum L_i \log(S_i) = - 1 \cdot \log(0.05) = 3$$

Redes Neuronales Convolucionales (CNN)

- Por tanto, la arquitectura consiste en capas anidadas de convoluciones+pooling, y finalmente capas interconectadas (“fully connected”)
- El tipo de arquitectura variará según el tipo de problema



Redes Neuronales Convolucionales (CNN)

- Ejemplo: ImageNet (concurso de reconocimiento de imágenes)
- Arquitectura ganadora (2012):

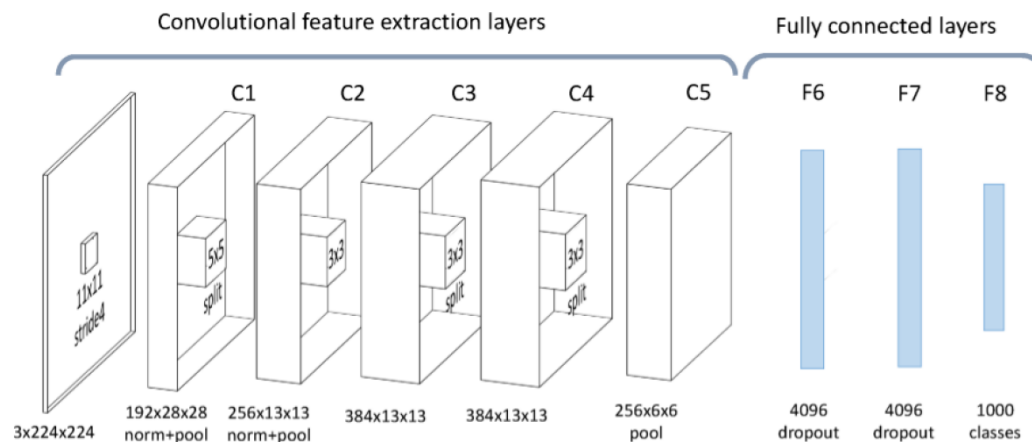
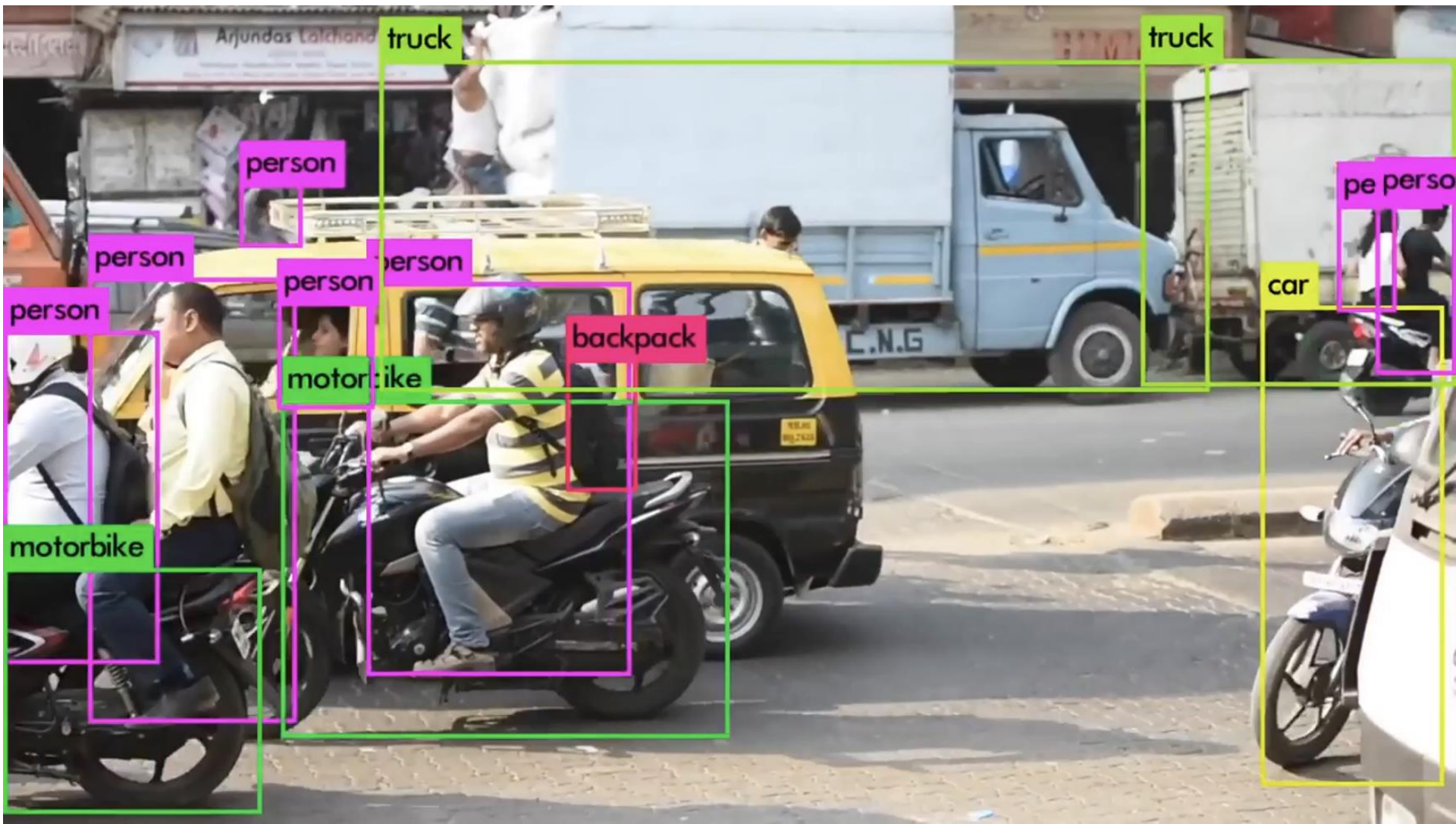


Fig. 2.3: Architecture for image recognition. The 2012 ILSVRC winner consists of eight layers [82]. Each layer performs a linear transformation (specifically, convolutions in layers C1–C5 and matrix multiplication in layers F6–F8) followed by nonlinear transformations (rectification in all layers, contrast normalization in C1–C2, and pooling in C1–C2 and C5). Regularization with dropout noise is used in layers F6–F7.

Redes Neuronales Convolucionales (CNN)



Redes Neuronales Convolucionales (CNN)

Algoritmo You Only Look Once (YOLO)

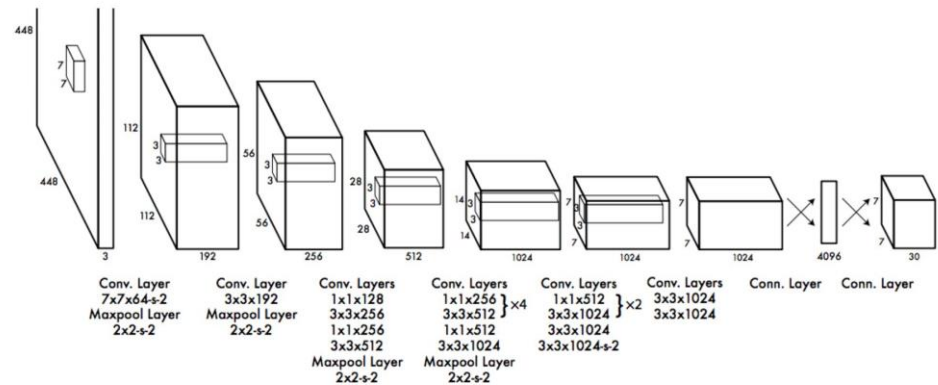
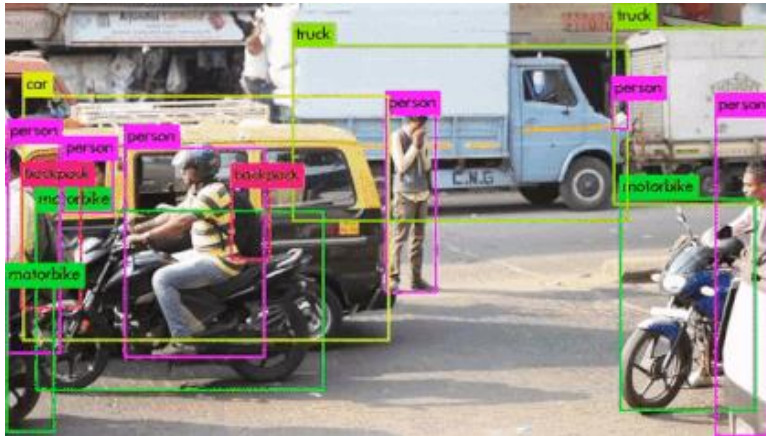


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Redes Neuronales Convolucionales (CNN)



"man in black shirt is playing guitar."



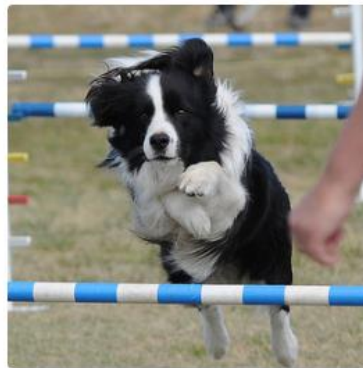
"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."



"girl in pink dress is jumping in air."

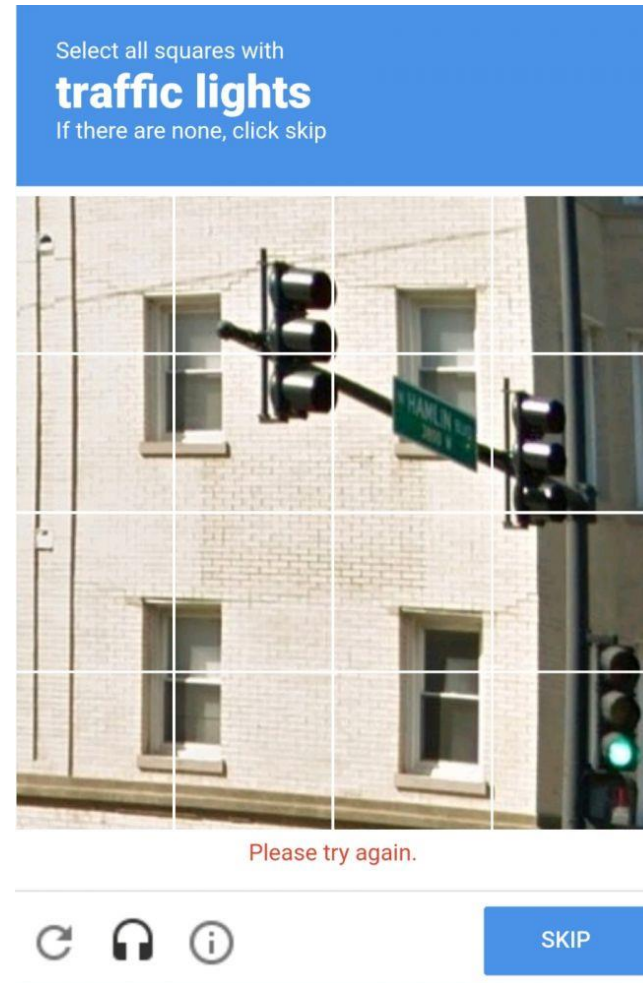
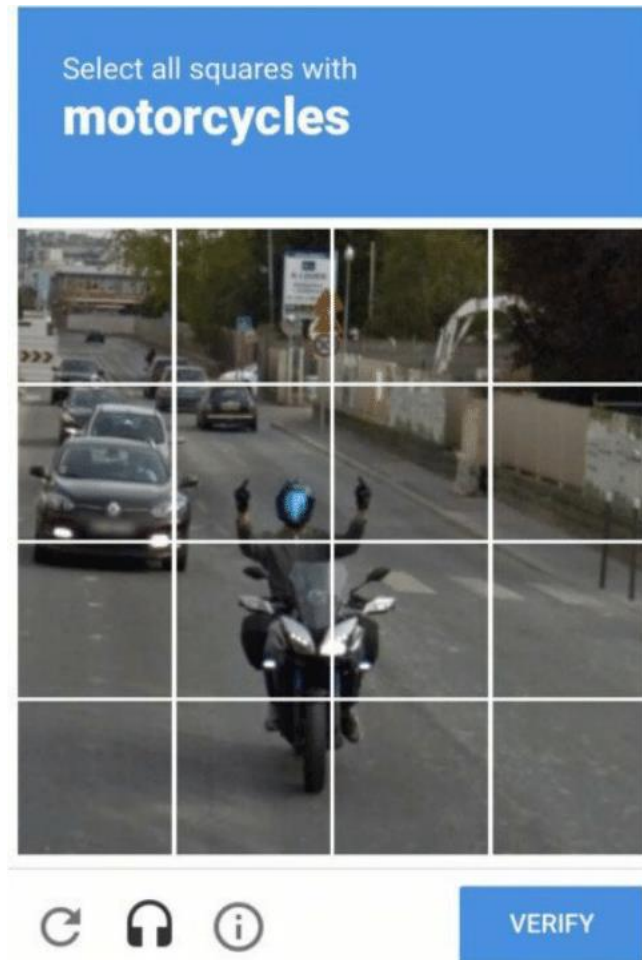


"black and white dog jumps over bar."



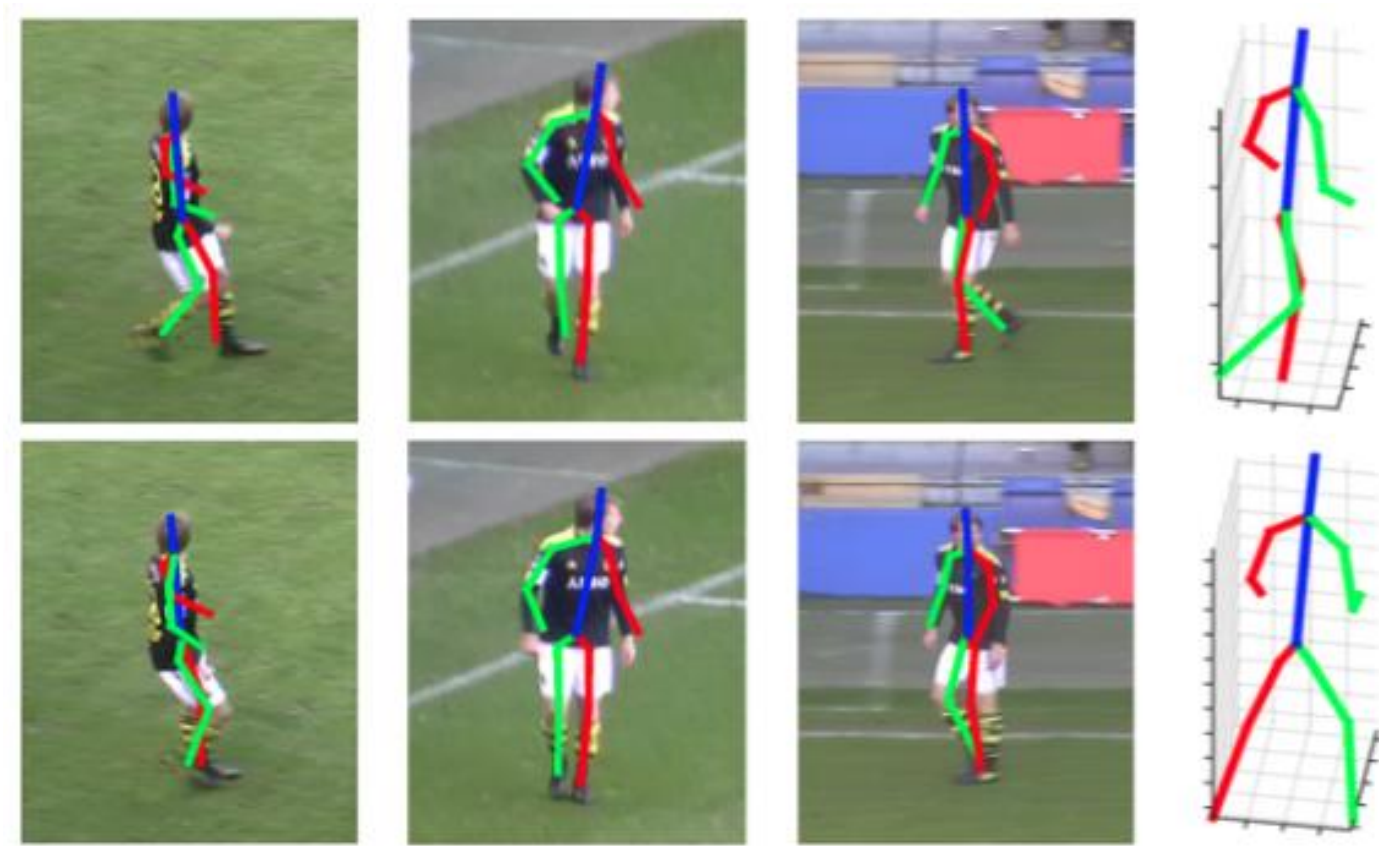
"young girl in pink shirt is swinging on swing."

Redes Neuronales Convolucionales (CNN)

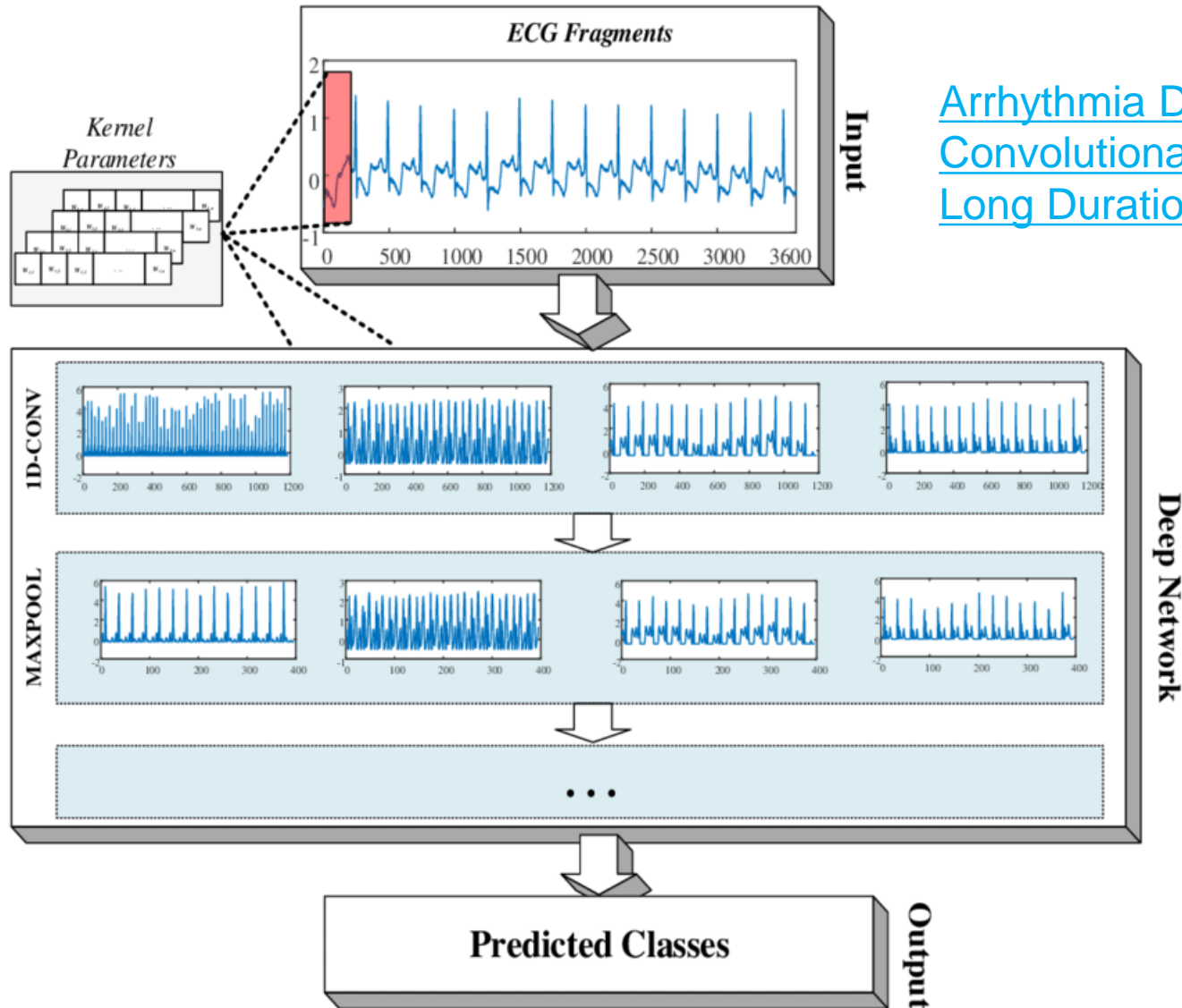


Caso de uso: App del Jugador (LaLiga)

3. Detección de pose



No solo en imágenes...

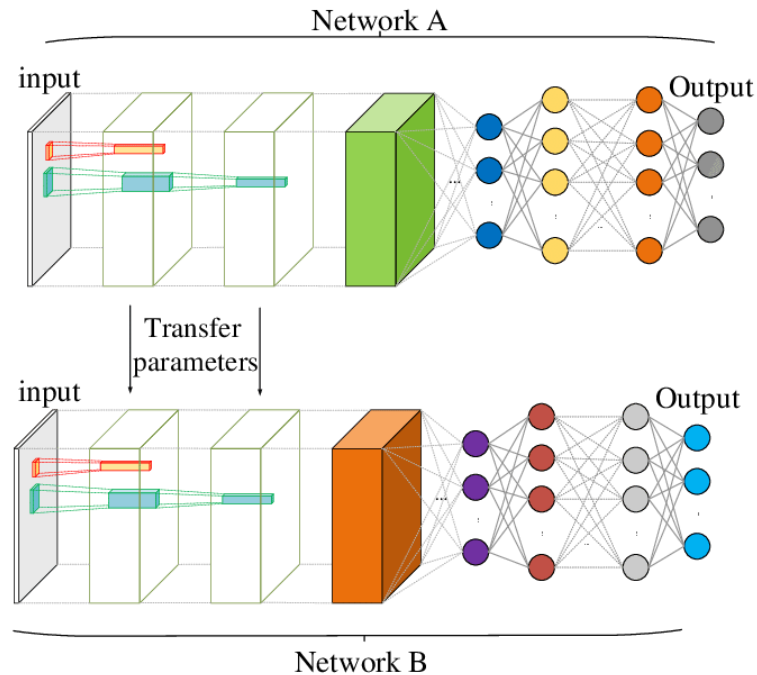


[Arrhythmia Detection Using Deep Convolutional Neural Network With Long Duration ECG Signals](#)

Transfer Learning



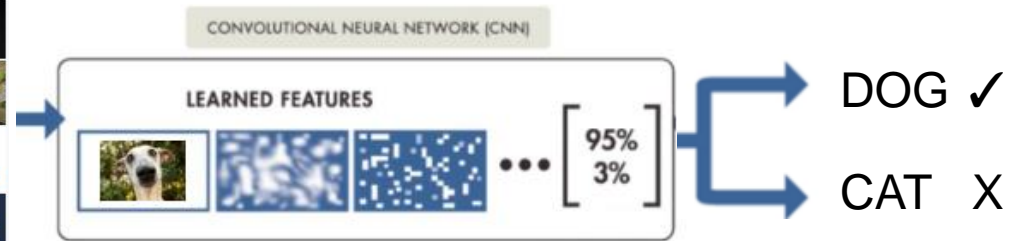
- Si no tenemos suficientes ejemplos de entrenamiento, podemos utilizar una red pre-entrenada y cambiarle las etiquetas salida.
- Las características de bajo nivel extraídas por la primera red, pueden servir para la nueva tarea



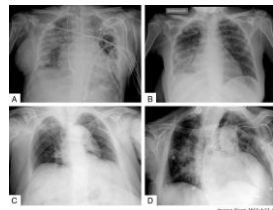
Transfer Learning



1 MILLÓN DE IMÁGENES

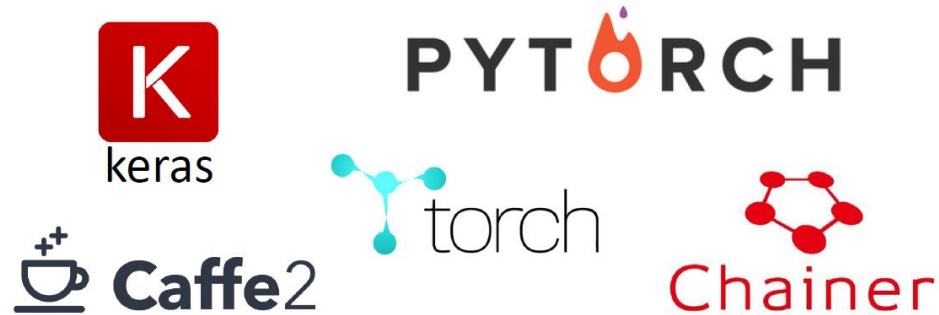


100 IMÁGENES



Deep Learning frameworks

- Alto nivel



- Bajo nivel



Lectura recomendada

- **How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras**

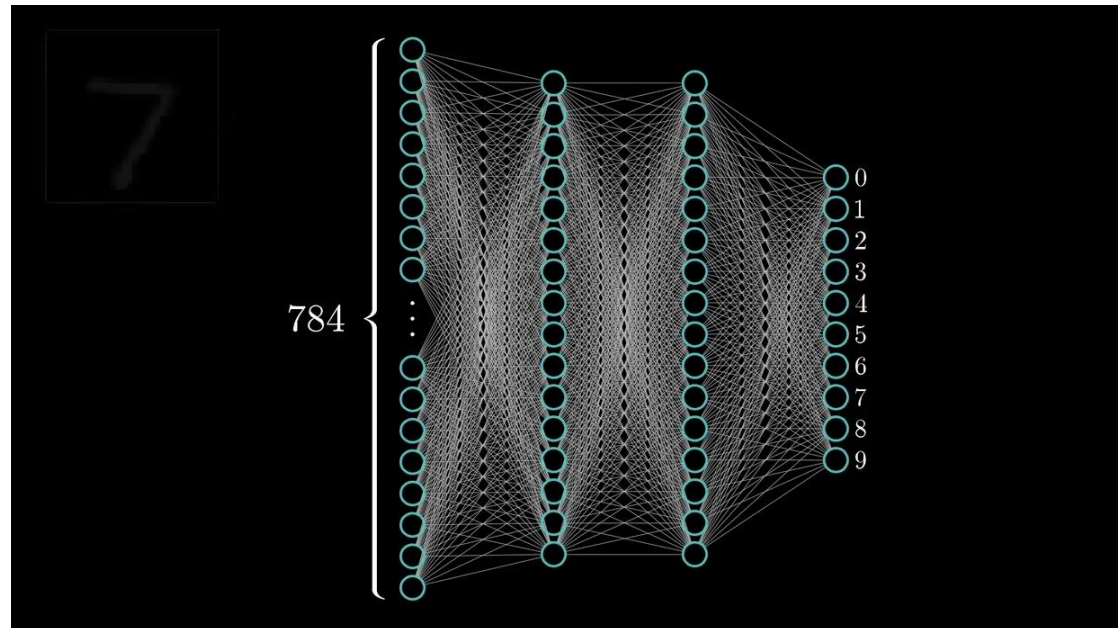
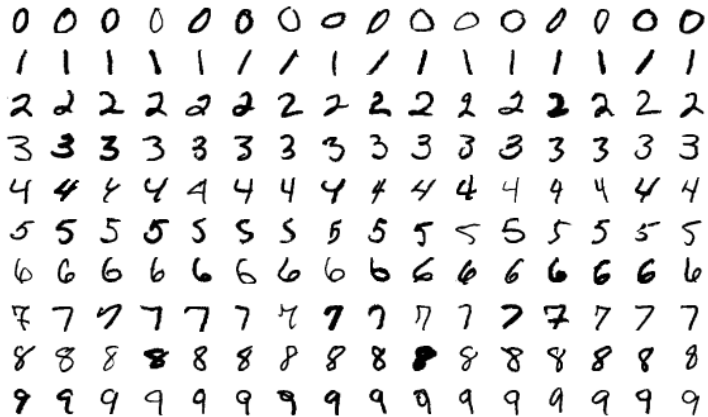
<https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/>



Práctica

Reconocimiento de dígitos manuscritos

- El dataset [MNIST](#) contiene 70.000 imágenes de tamaño 28x28 píxeles con dígitos del 0 al 9 escritos a mano
- Utilizaremos la librería [keras](#) de Python para crear clasificadores basados en redes neuronales
- [Enlace al notebook de Python](#)



Do it yourself

Reconocimiento de imágenes

- El conjunto de datos CIFAR-10 consta de 60000 imágenes en color de 32x32 con 10 clases, y 6000 imágenes por clase. Hay 50000 imágenes de entrenamiento y 10000 imágenes de prueba.
- [Enlace al notebook de Python](#)

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck

