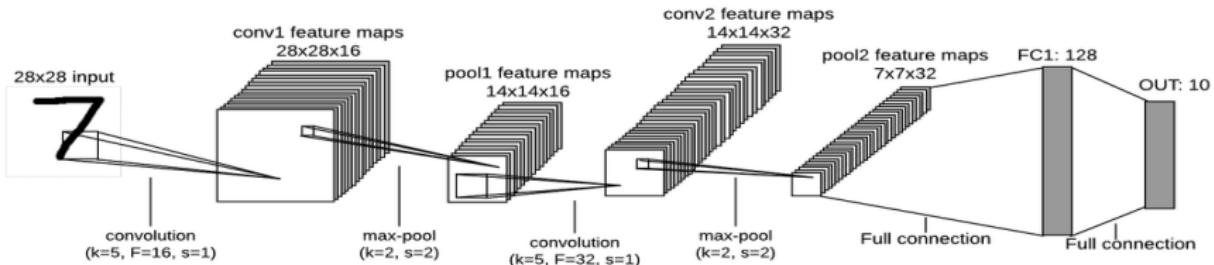


Redes convolucionales

nociónes básicas para su aplicación

Julio Waissman



Plan de la presentación

- ① Información local vs información global
- ② ¿Cómo son las redes neuronales convolucionales?
- ③ ¿Cómo funciona esto?
- ④ Consideraciones finales

¿Qué es el reconocimiento de imágenes



Imagen

Imagen original



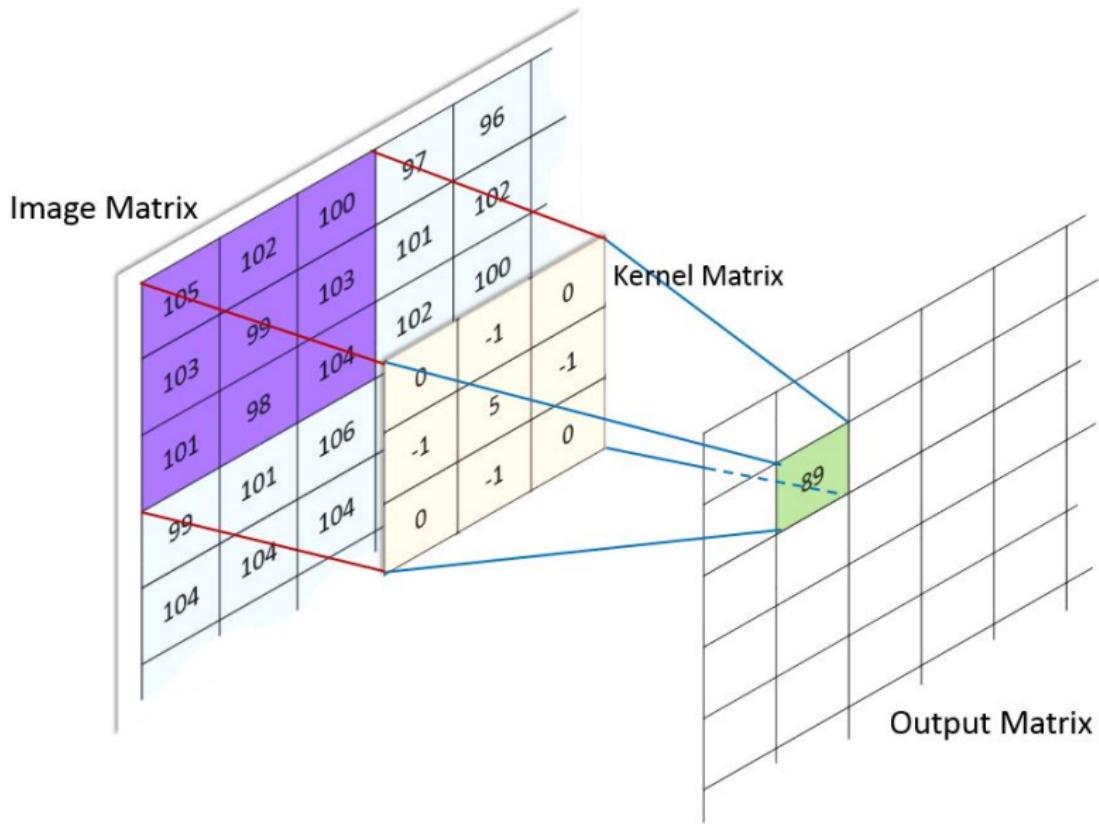
Imagen como matriz de números

Detalle

105	102	100	97	96	
103	99	103	101	102	
101	98	104	102	100	
99	101	106	104	99	
104	104	104	100	98	



Filtro convolucional



Filtro convolucional

105	102	100	97	96	
103	99	103	101	102	
101	98	104	102	100	
99	101	106	104	99	
104	104	104	100	98	

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

		89		

Output Matrix

$$\begin{aligned} & 105 * 0 + 102 * -1 + 100 * 0 \\ & + 103 * -1 + 99 * 5 + 103 * -1 \\ & + 101 * 0 + 98 * -1 + 104 * 0 = 89 \end{aligned}$$

Filtro convolucional

105	102	100	97	96	
103	99	103	101	102	
101	98	104	102	100	
99	101	106	104	99	
104	104	104	100	98	

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

		89	111	

Output Matrix

$$\begin{aligned} & 102 * 0 + 100 * -1 + 97 * 0 \\ & + 99 * -1 + 103 * 5 + 101 * -1 \\ & + 98 * 0 + 104 * -1 + 102 * 0 = 111 \end{aligned}$$

Filtro convolucional

0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

210	89	111		

Output Matrix

$$\begin{aligned} & 0 * 0 + 105 * -1 + 102 * 0 \\ & + 0 * -1 + 103 * 5 + 99 * -1 \\ & + 0 * 0 + 101 * -1 + 98 * 0 = 210 \end{aligned}$$

Filtro convolucional

0	0	0	0	0	0	
0	105	102	100	97	96	
0	103	99	103	101	102	
0	101	98	104	102	100	
0	99	101	106	104	99	
0	104	104	104	100	98	

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

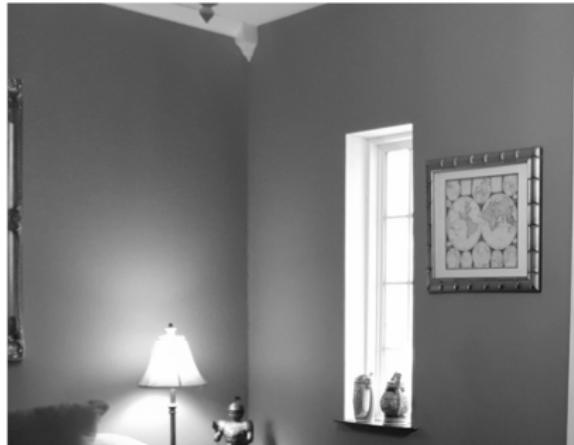
320				
210	89	111		

Output Matrix

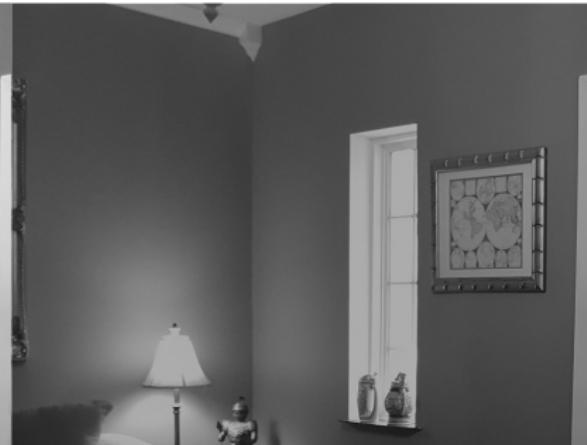
$$\begin{aligned} & 0 * 0 + 0 * -1 + 0 * 0 \\ & + 0 * -1 + 105 * 5 + 102 * -1 \\ & + 0 * 0 + 103 * -1 + 99 * 0 = 320 \end{aligned}$$

Resultado

Original



Filtrada



Los filtros pueden ser extremos

$$Kernel = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Y el procesamiento de imágenes de hace así...

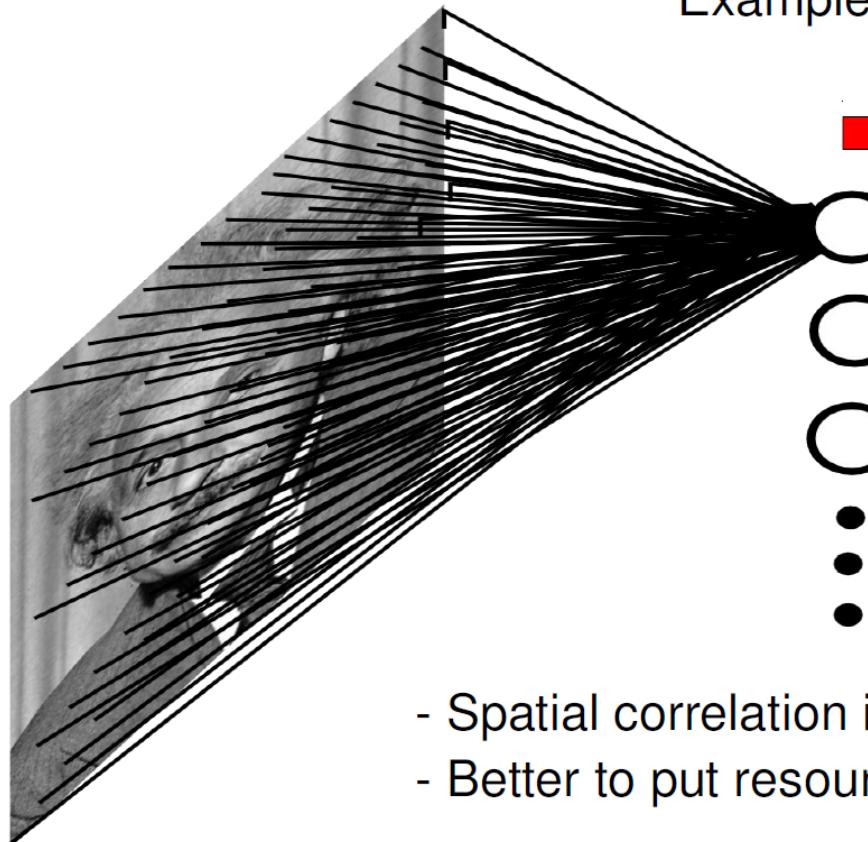
- Primero se seleccionan muchos filtros, dependiendo de lo que queremos
- Se ajustan los filtros para representar las características principales
- Se pueden aplicar filtros a las imágenes filtradas
- Se utilizan otras técnicas (submuestreo, histograma, ...)
- Se convierte la imagen en un enorme vector de características

¿Y porqué nos interesaría el reconocimiento de imágenes?

- Una imagen es información organizada en varias dimensiones
- La información temporal se puede organizar en dimensiones (días/años, etc.)
- La información adyacente a un dato se asume más importante que la información más lejana
- Muchos problemas se pueden representar de esta forma (¿Cómo podrían representarse como imágenes los datos de demanda de energía?)

CNNs – key ideas

FULLY CONNECTED NEURAL NET

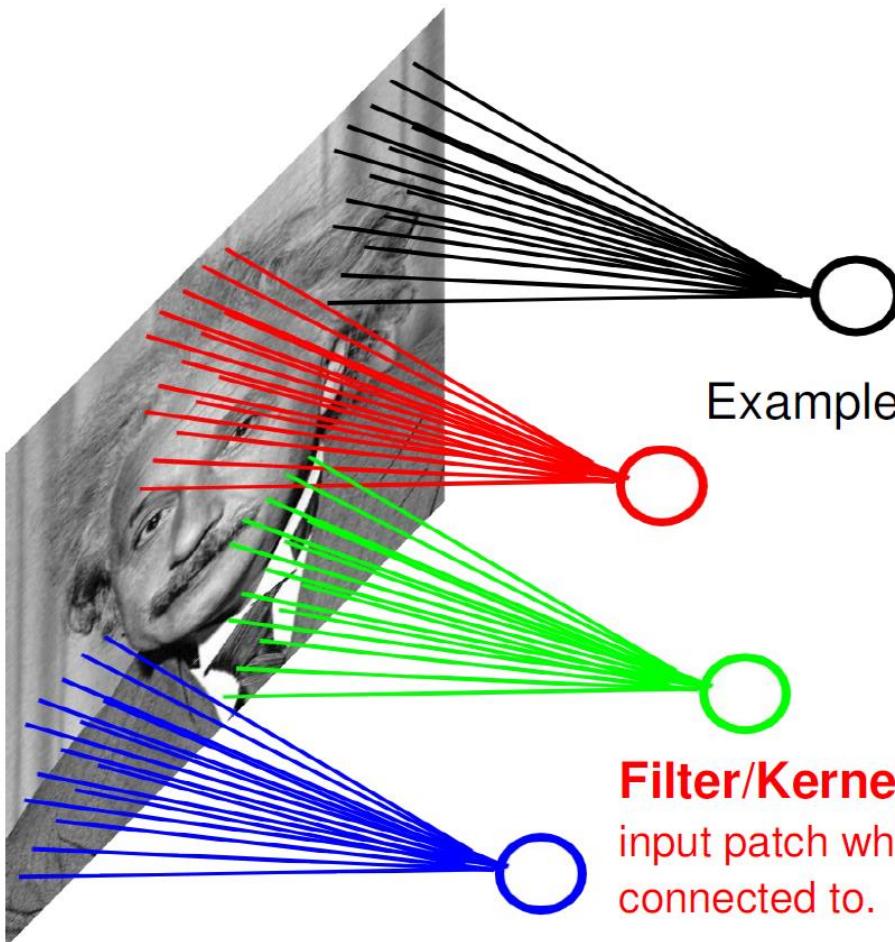


Example: 1000x1000 image
1M hidden units
→ **10¹² parameters!!!**

- Spatial correlation is local
- Better to put resources elsewhere!

CNNs – key ideas

LOCALLY CONNECTED NEURAL NET

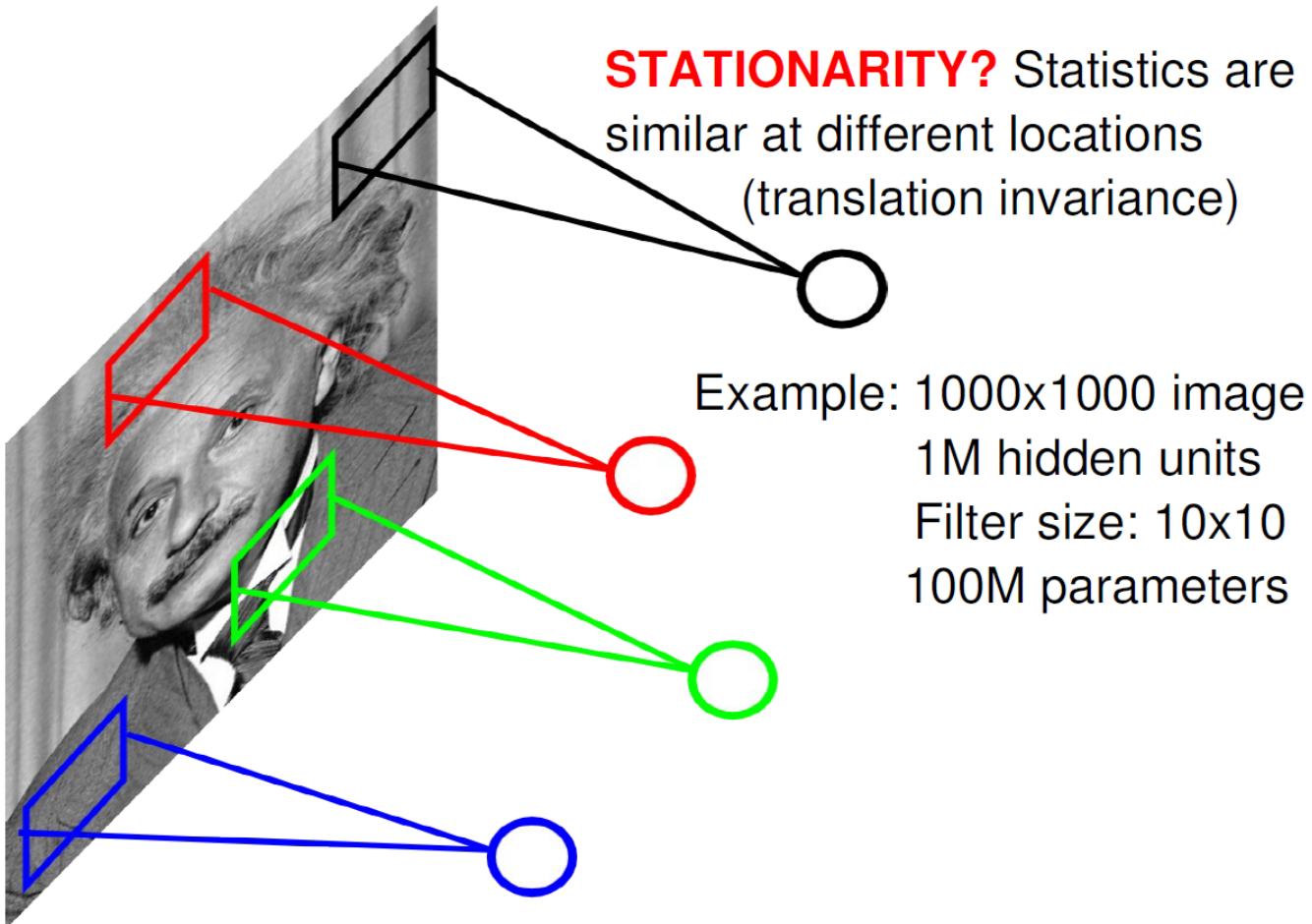


Example: 1000x1000 image
1M hidden units
Filter size: 10x10
100M parameters

Filter/Kernel/Receptive field:
input patch which the hidden unit is
connected to.

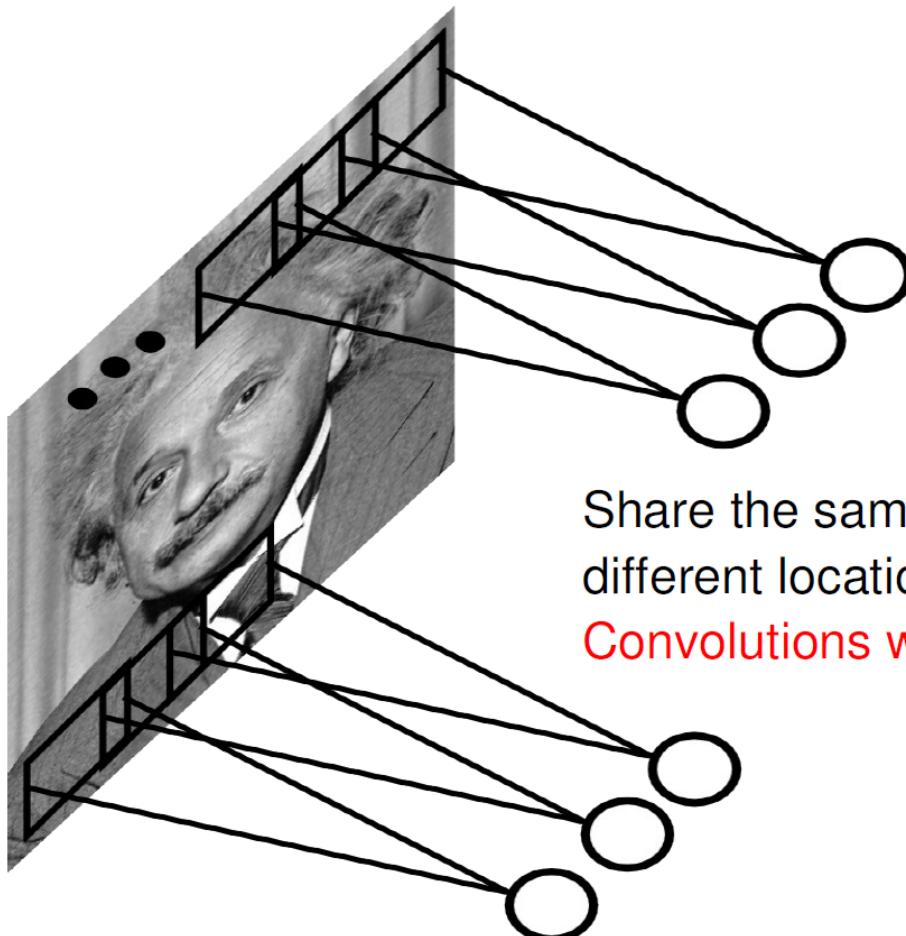
CNNs – key ideas

LOCALLY CONNECTED NEURAL NET



CNNs – key ideas

CONVOLUTIONAL NET

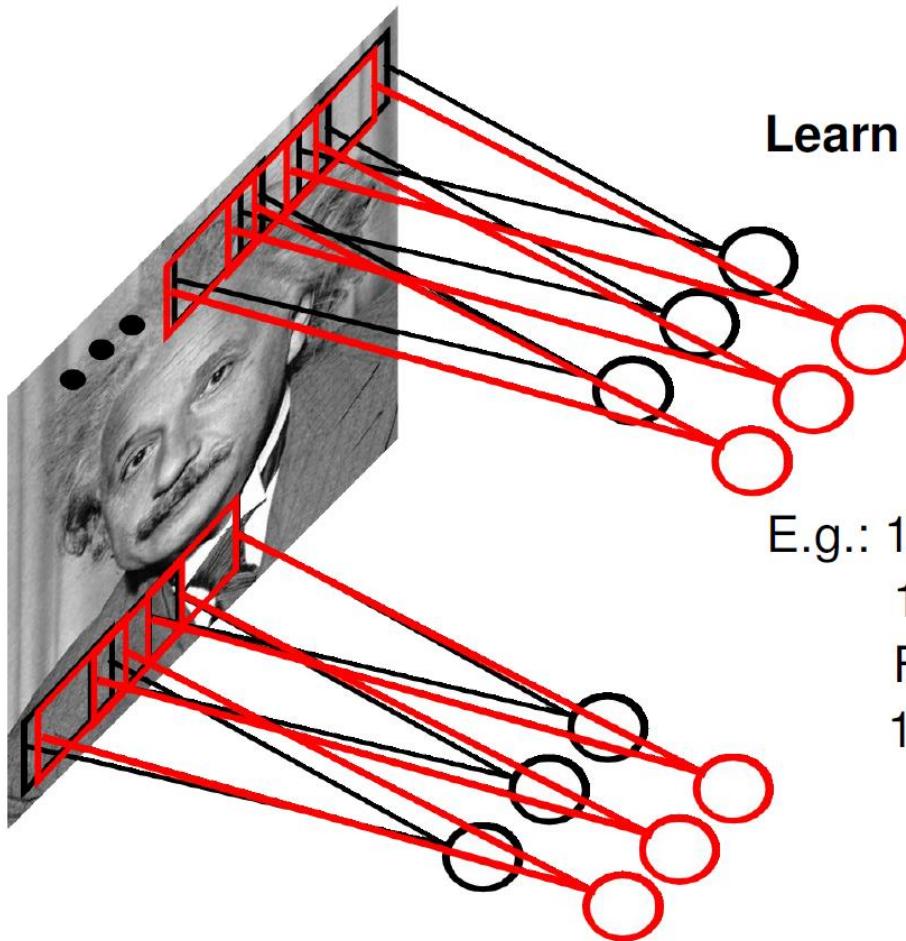


Share the same parameters across
different locations:

Convolutions with learned kernels

CNNs – key ideas

CONVOLUTIONAL NET

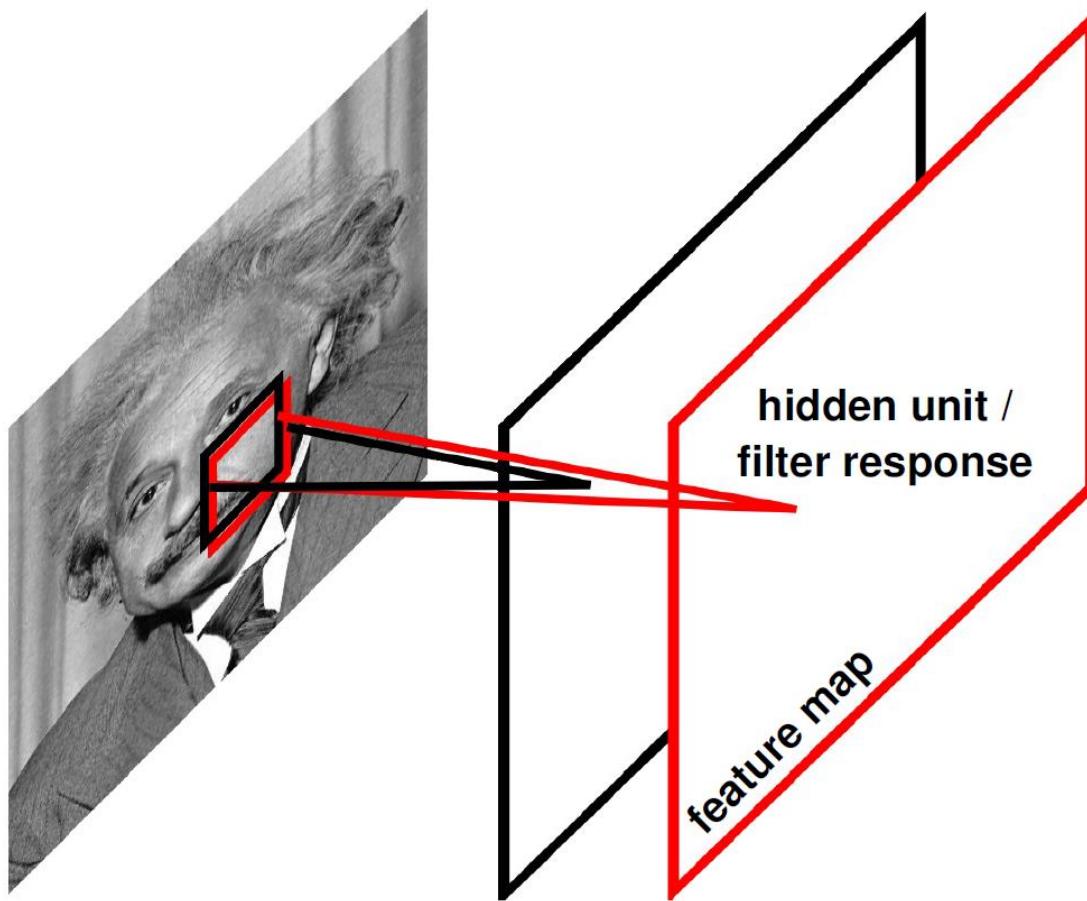


Learn multiple filters.

E.g.: 1000x1000 image
100 Filters
Filter size: 10x10
10K parameters

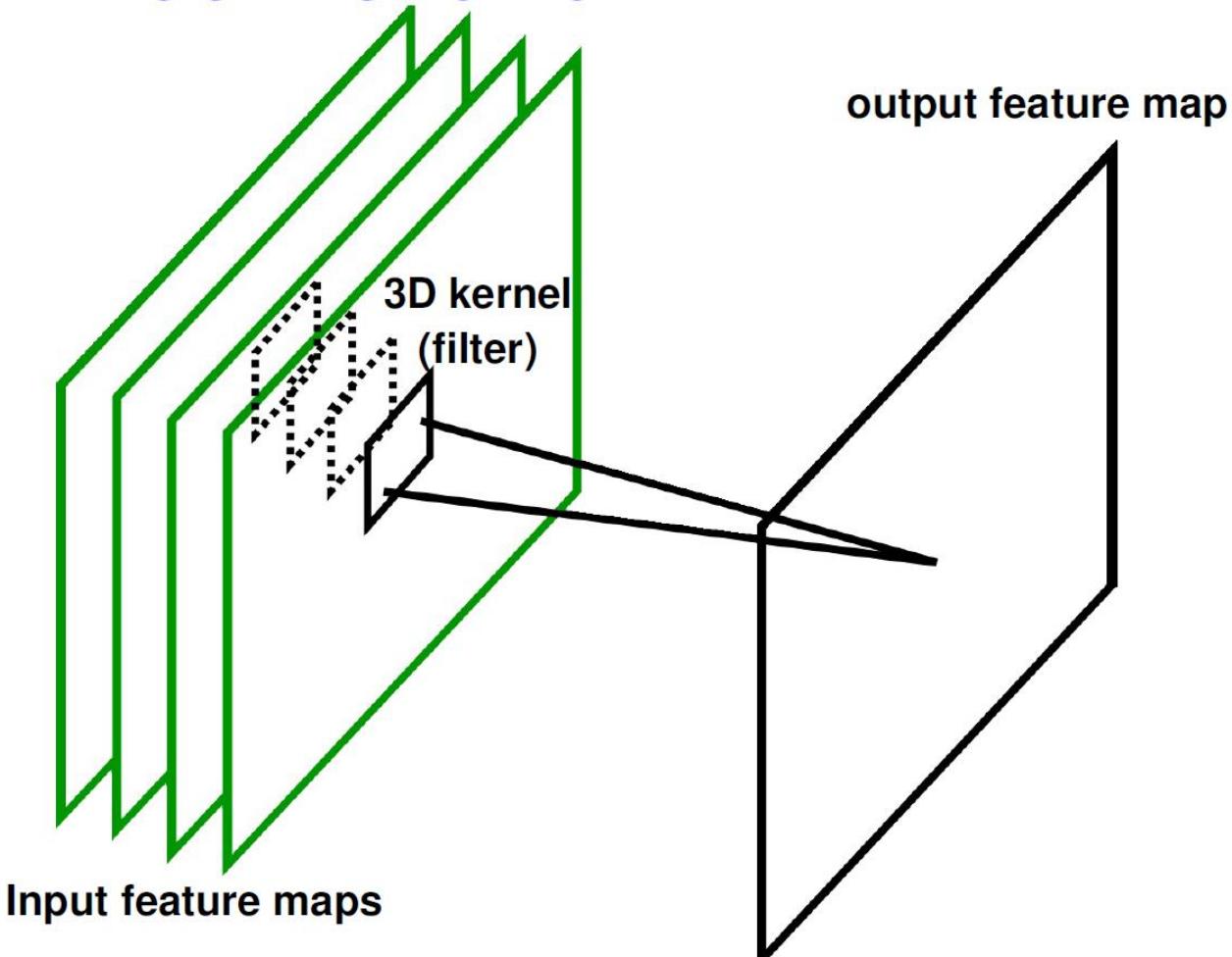
CNNs – key ideas

CONVOLUTIONAL NET



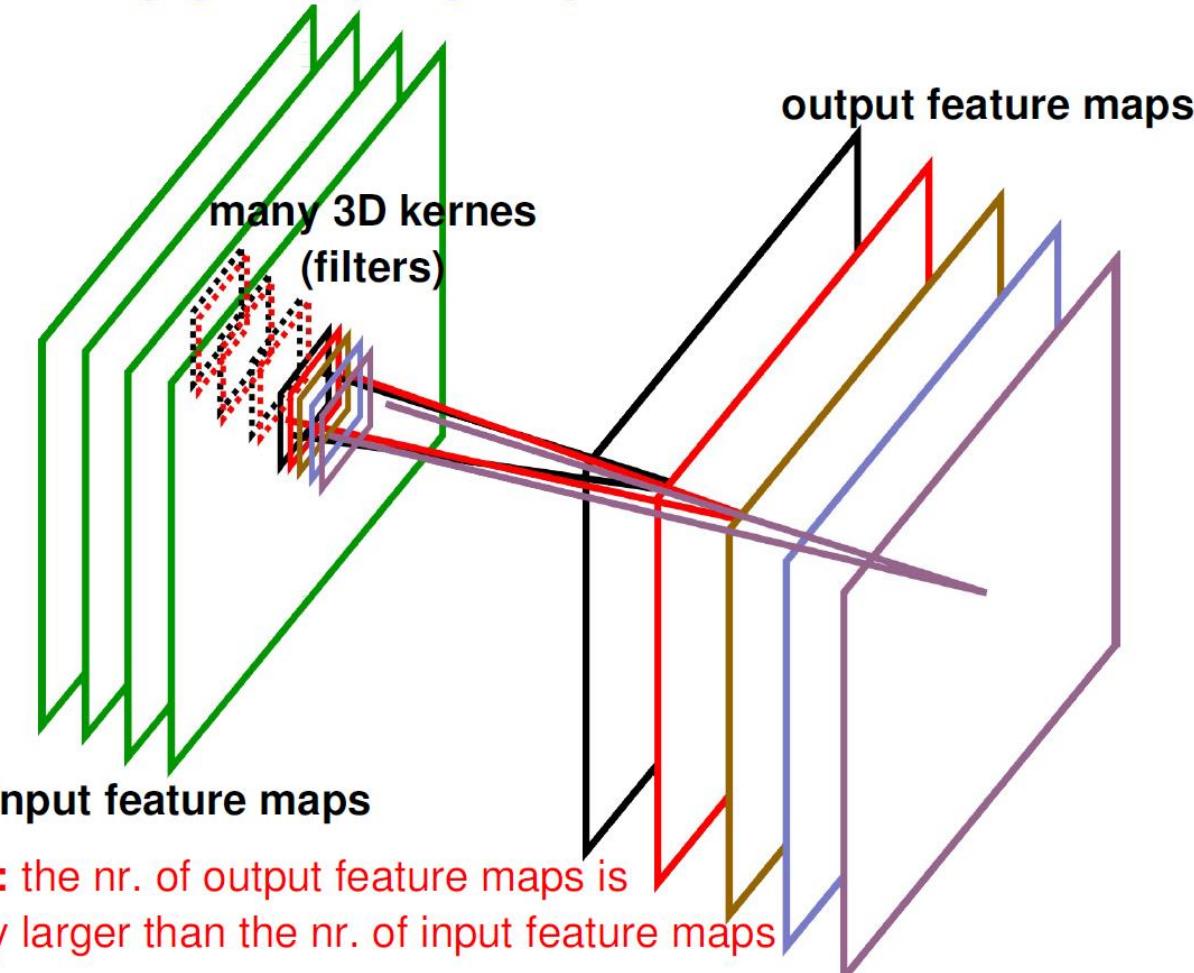
CNNs – key ideas

CONVOLUTIONAL LAYER



CNNs – key ideas

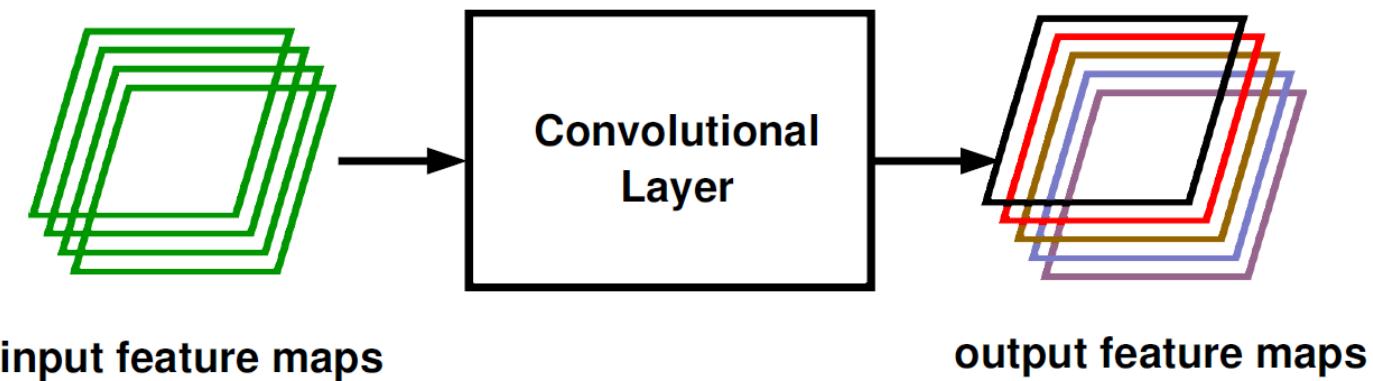
CONVOLUTIONAL LAYER



NOTE: the nr. of output feature maps is
usually larger than the nr. of input feature maps

CNNs – key ideas

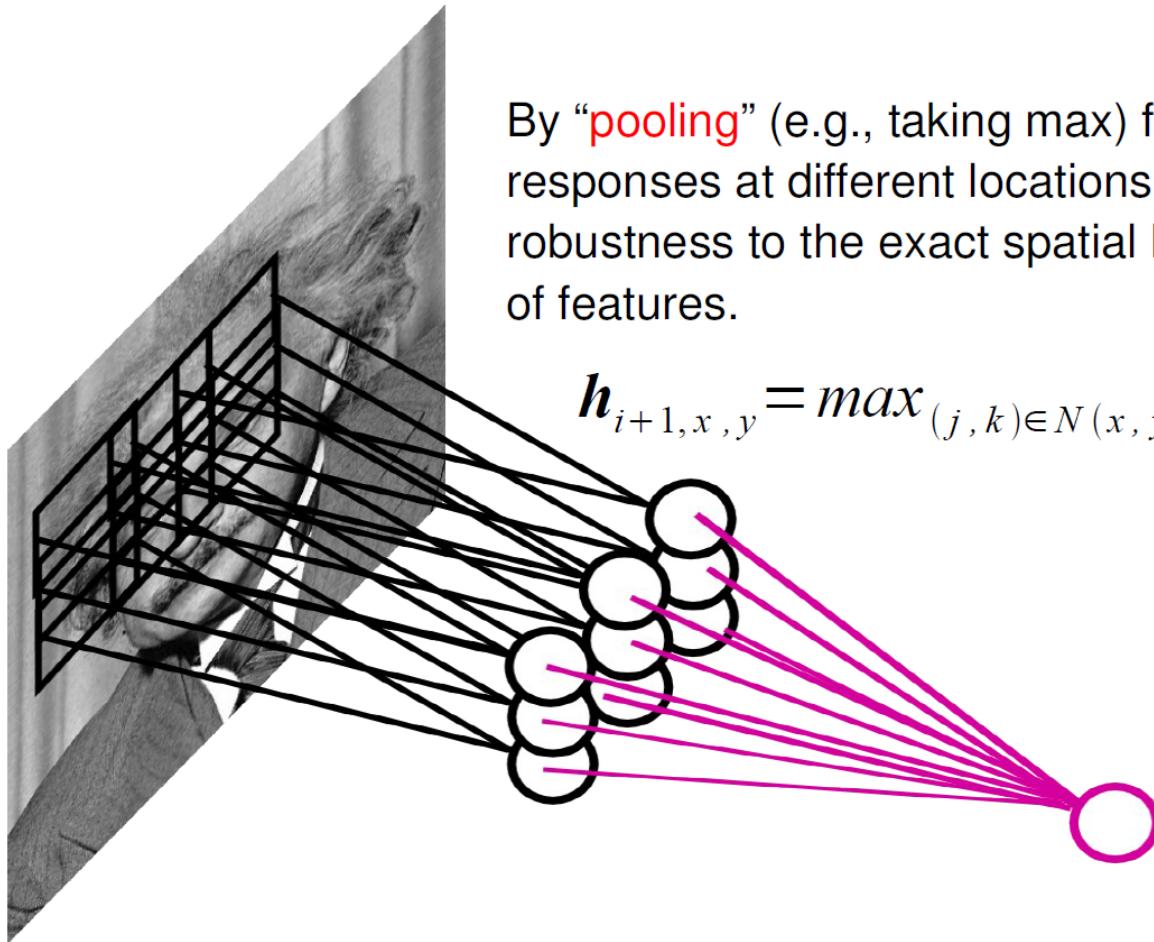
CONVOLUTIONAL LAYER



NOTE: the nr. of output feature maps is usually larger than the nr. of input feature maps

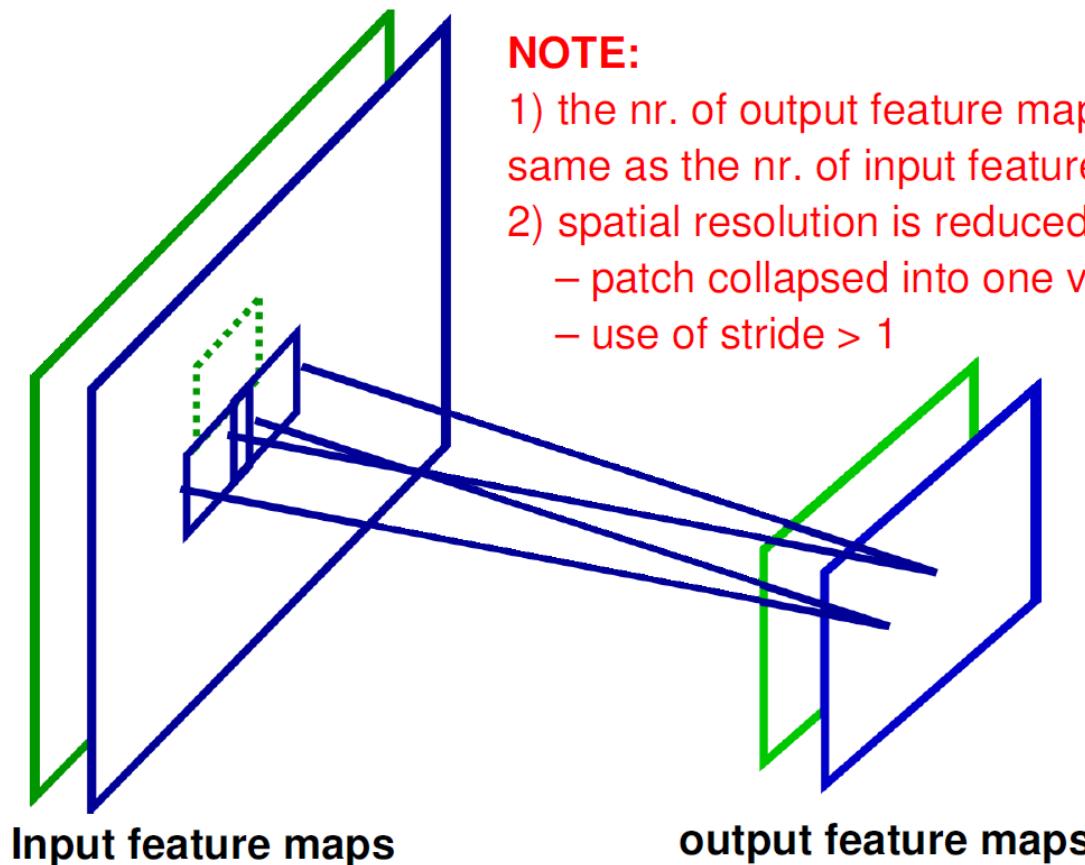
CNNs – key ideas

POOLING



CNNs – key ideas

POOLING LAYER

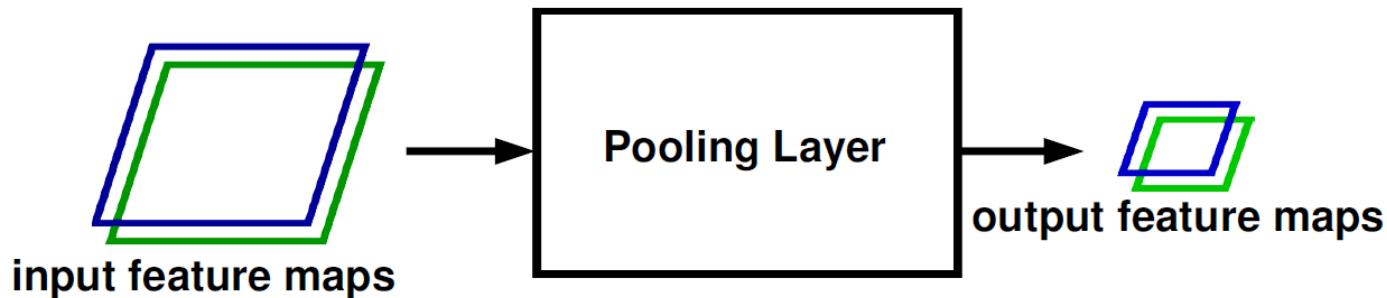


CNNs – key ideas

POOLING LAYER

NOTE:

- 1) the nr. of output feature maps is the same as the nr. of input feature maps
- 2) spatial resolution is reduced
 - patch collapsed into one value
 - use of stride > 1



CNNs – typical architecture

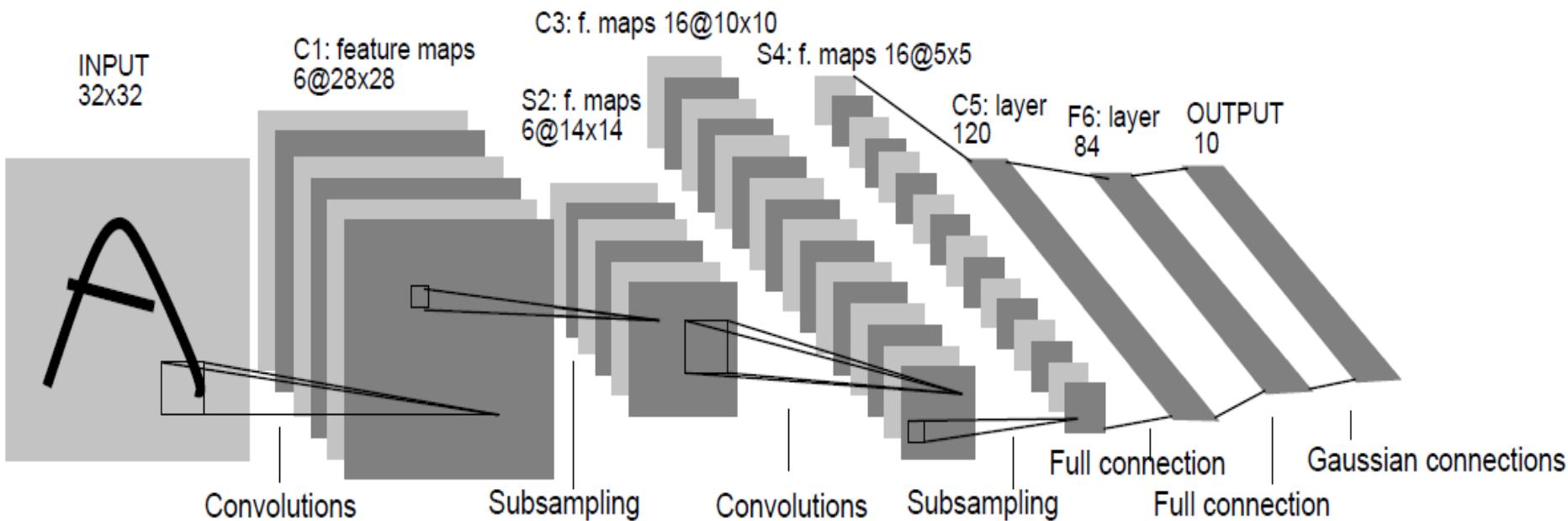
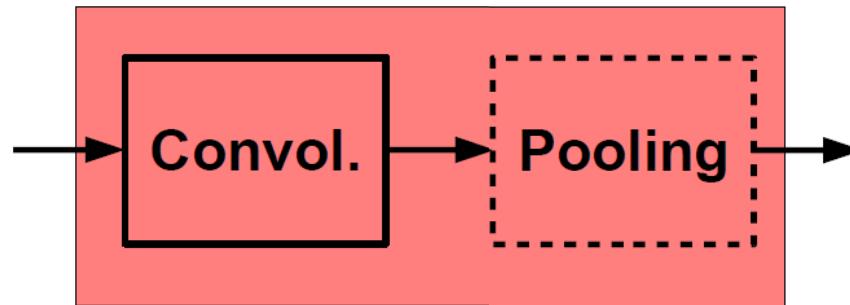


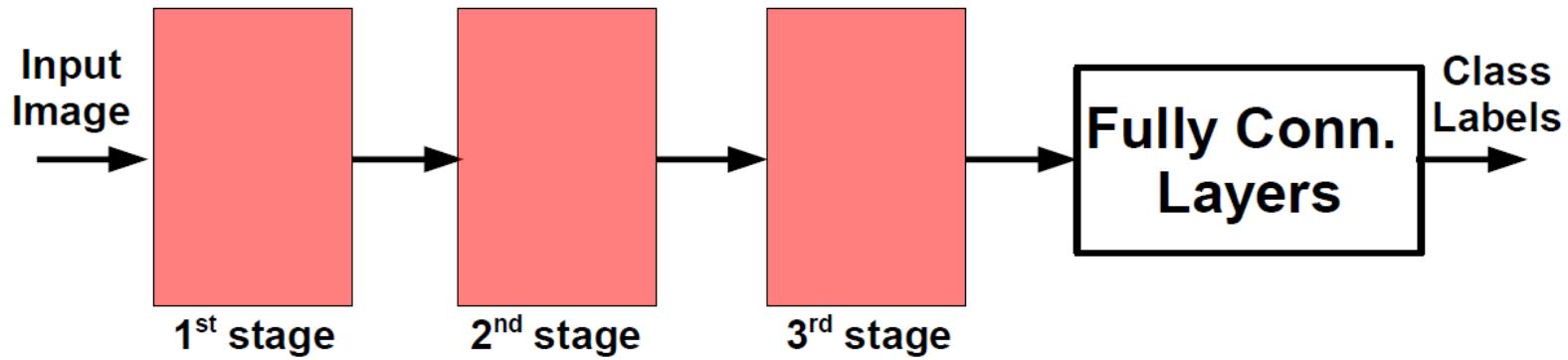
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

CNNs – typical architecture

One stage (zoom)



Whole system



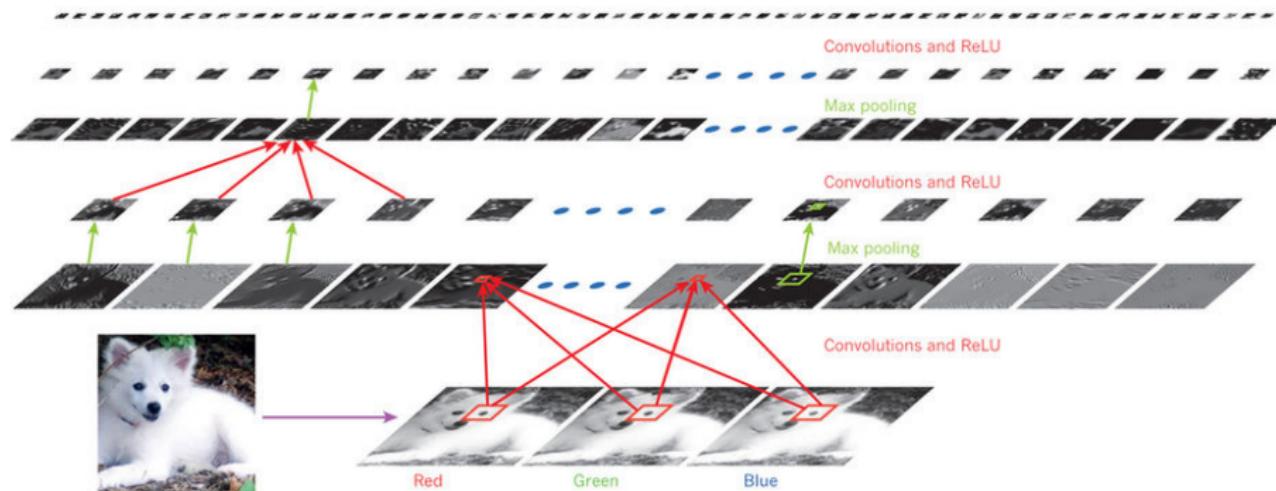
CNNs – conclusion

- Connect each hidden unit to a small patch of the input.
- Share the weight across hidden units.
- Subsampling layers are useful to reduce computational burden and increase invariance.

Redes convolucionales (CNN)

Arquitectura general

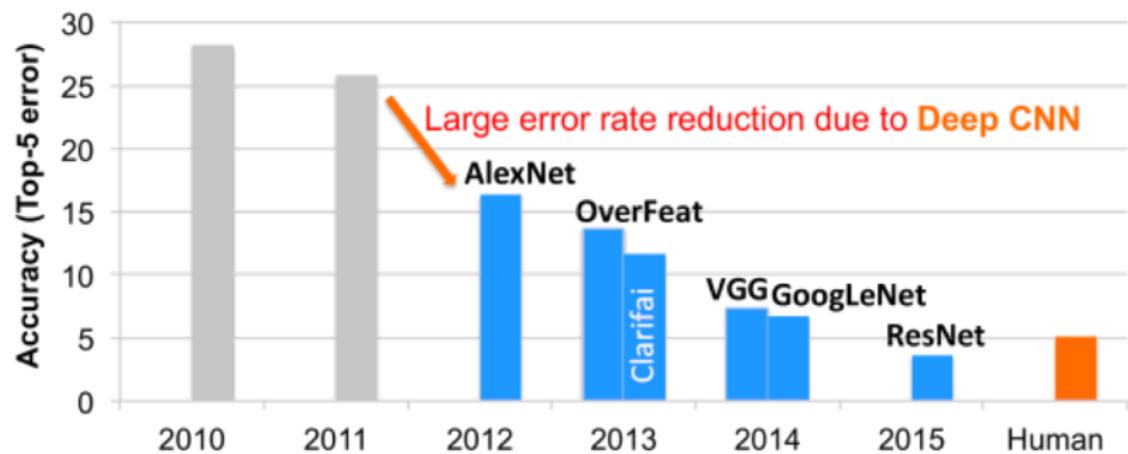
Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)



Redes convolucionales (CNN)

Desempeño en tratamiento de imágenes

Resultados aplicados al conjunto de datos *ImageNet*



... y colorín colorado ...

Muchas gracias por su atención