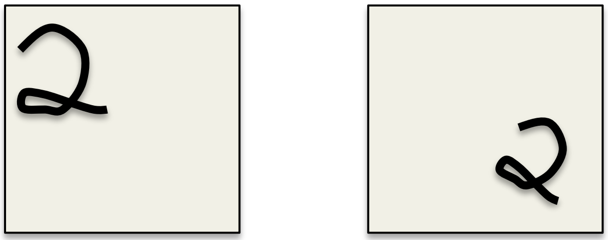
Things that make it hard to recognize objects

- Segmentation: Real scenes are cluttered with other objects:
  - Its hard to tell which pieces go together as parts of the same object.
  - Parts of an object can be hidden behind other objects.
- Lighting: The intensities of the pixels are determined as much by the lighting as by the objects.
- Deformation: Objects can deform in a variety of non-affine ways:
  - e.g a hand-written 2 can have a large loop or just a cusp.
- Affordances: Object classes are often defined by how they are used:
  - Chairs are things designed for sitting on so they have a wide variety of physical shapes.

More things that make it hard to recognize objects

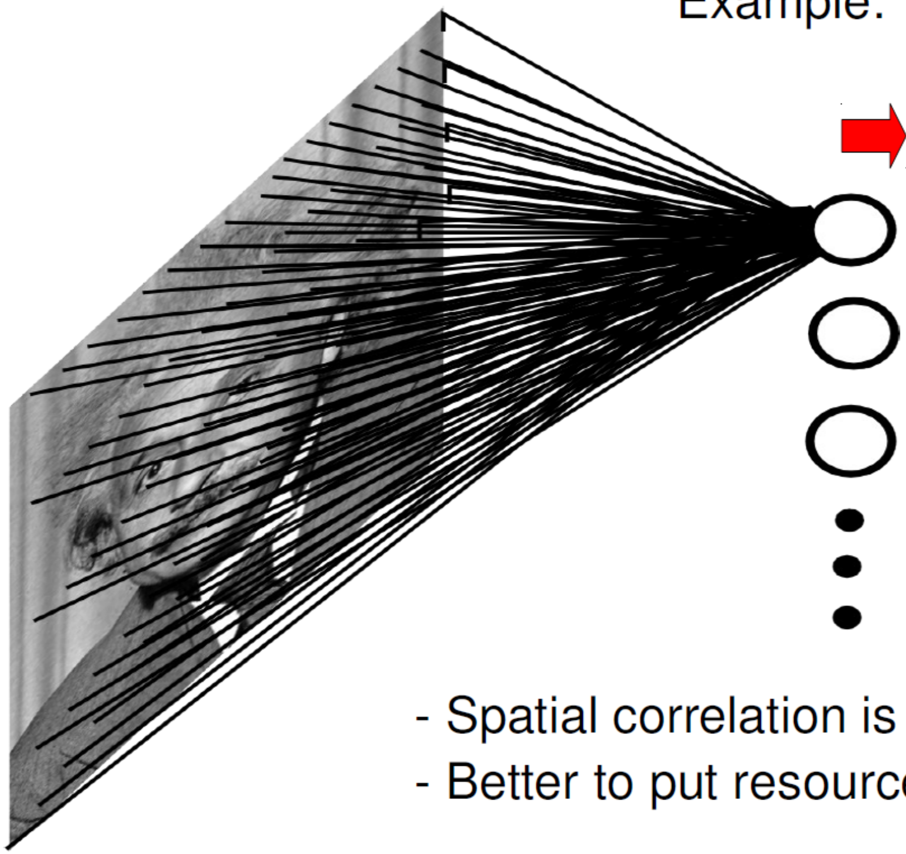
- Viewpoint: Changes in viewpoint cause changes in images that standard learning methods cannot cope with.
  - Information hops between input dimensions (*i.e.* pixels)



- Imagine a medical database in which the age of a patient sometimes hops to the input dimension that normally codes for weight!
  - To apply machine learning we would first want to eliminate this dimension-hopping.

CNNs – key ideas

FULLY CONNECTED NEURAL NET

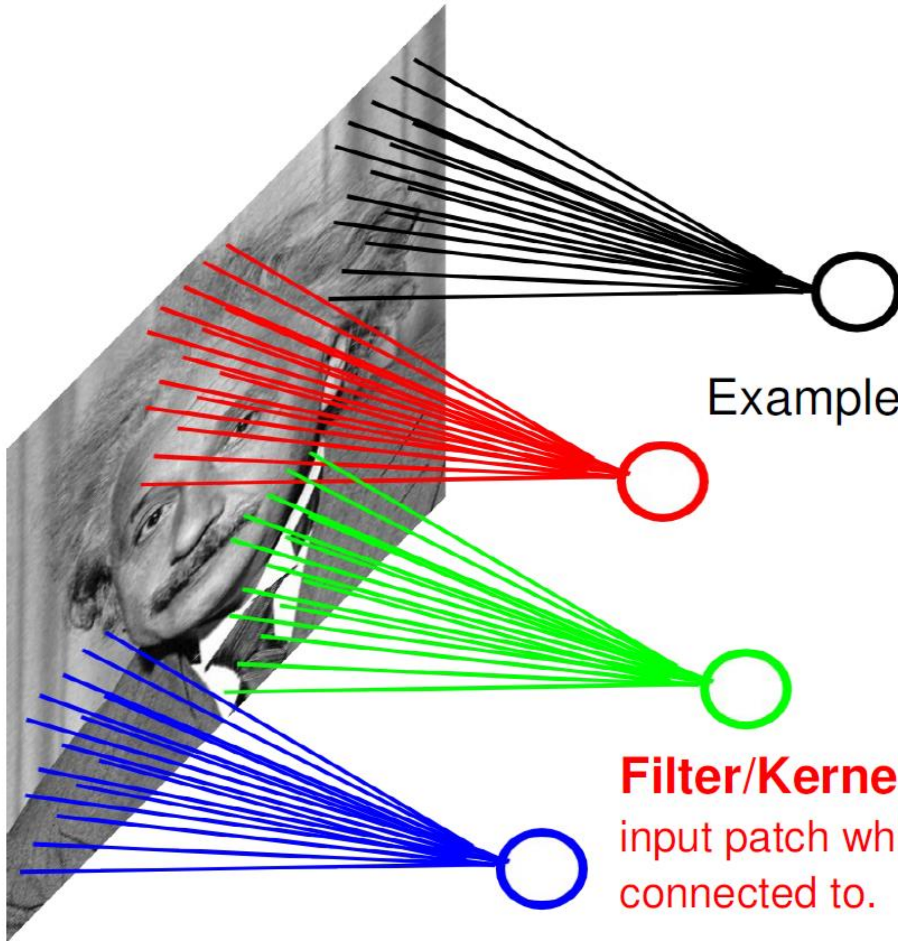


Example: 1000x1000 image
1M hidden units
10^12 parameters!!!

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

CNNs – key ideas

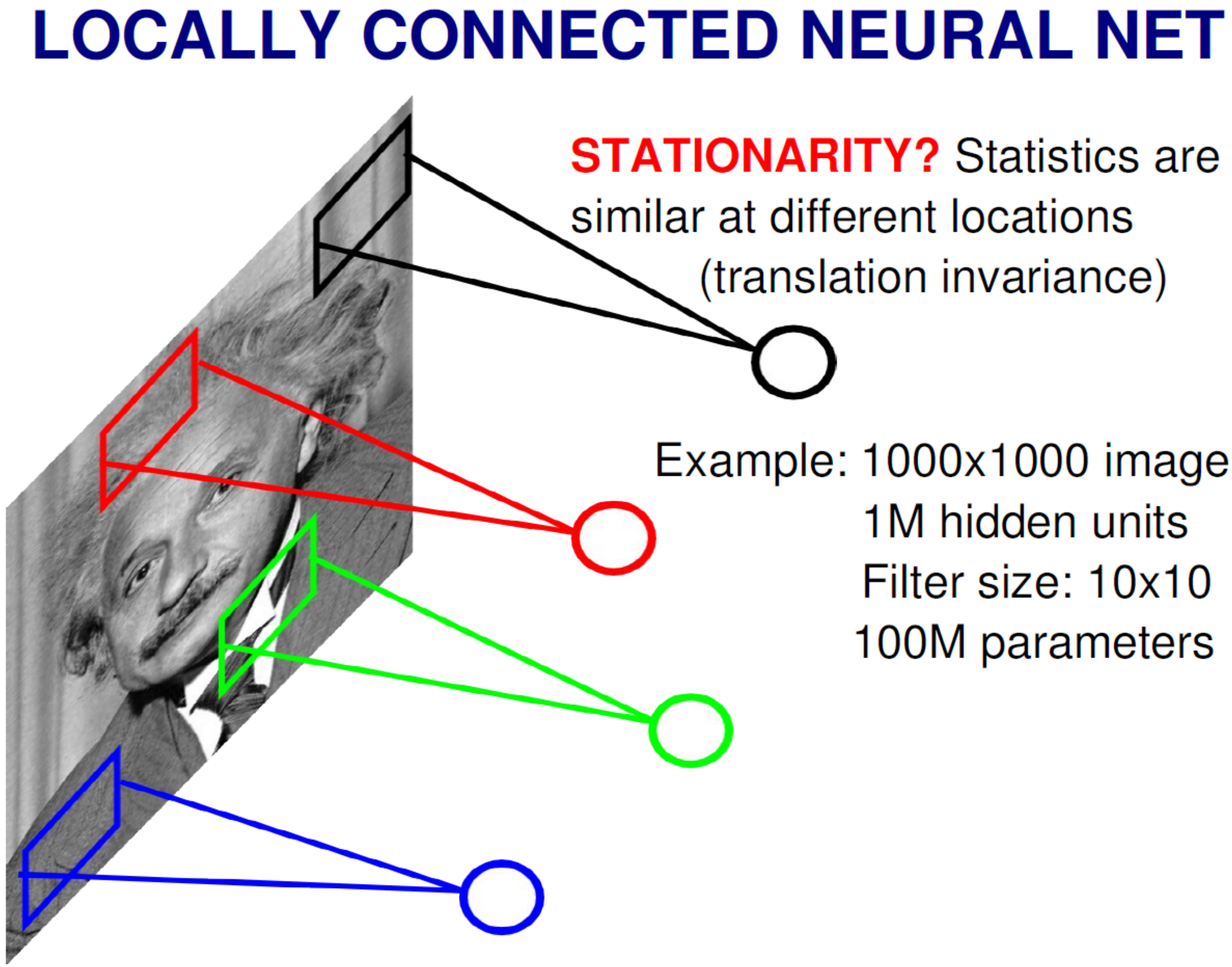
LOCALLY CONNECTED NEURAL NET



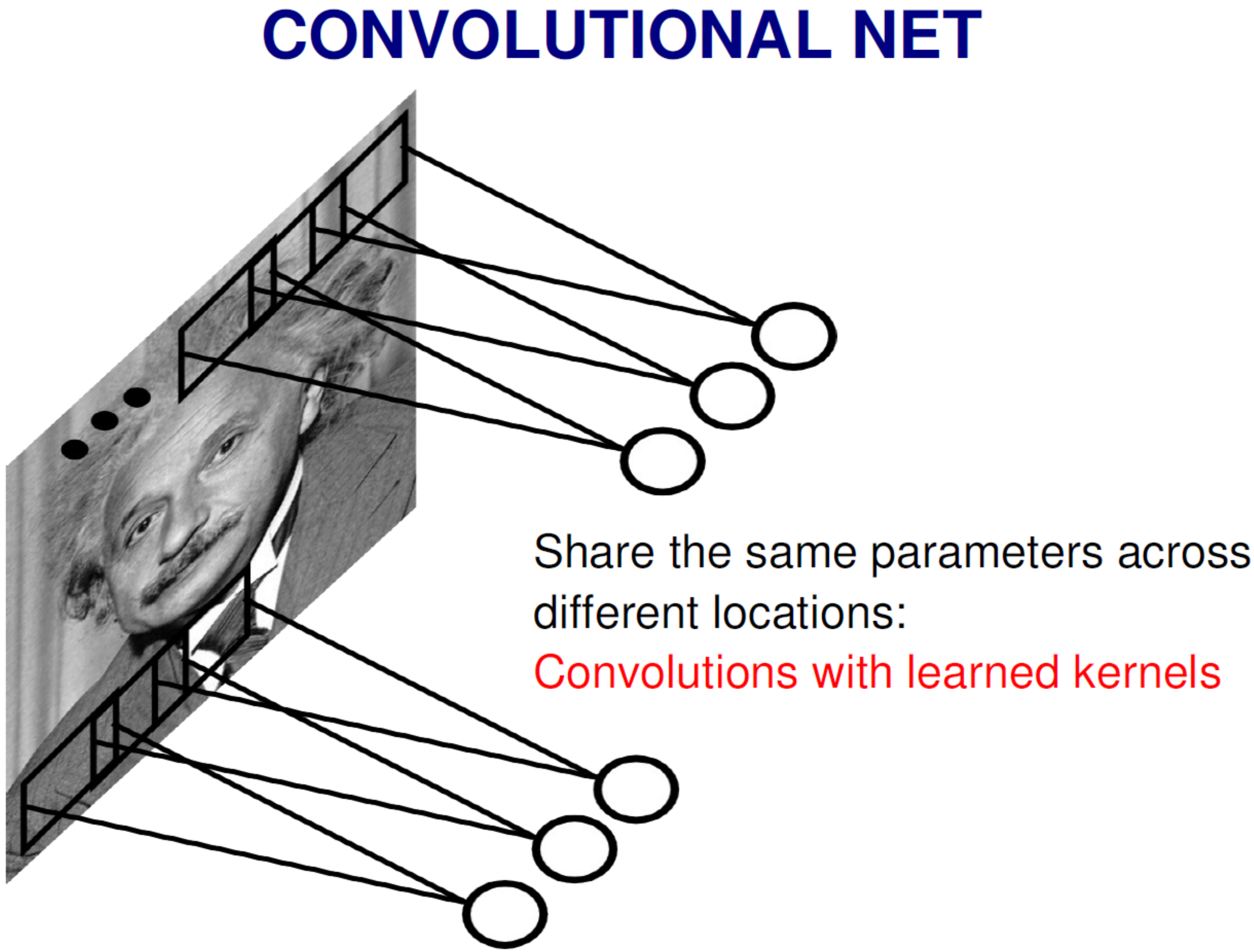
Example: 1000x1000 image
1M hidden units
Filter size: 10x10
100M parameters

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

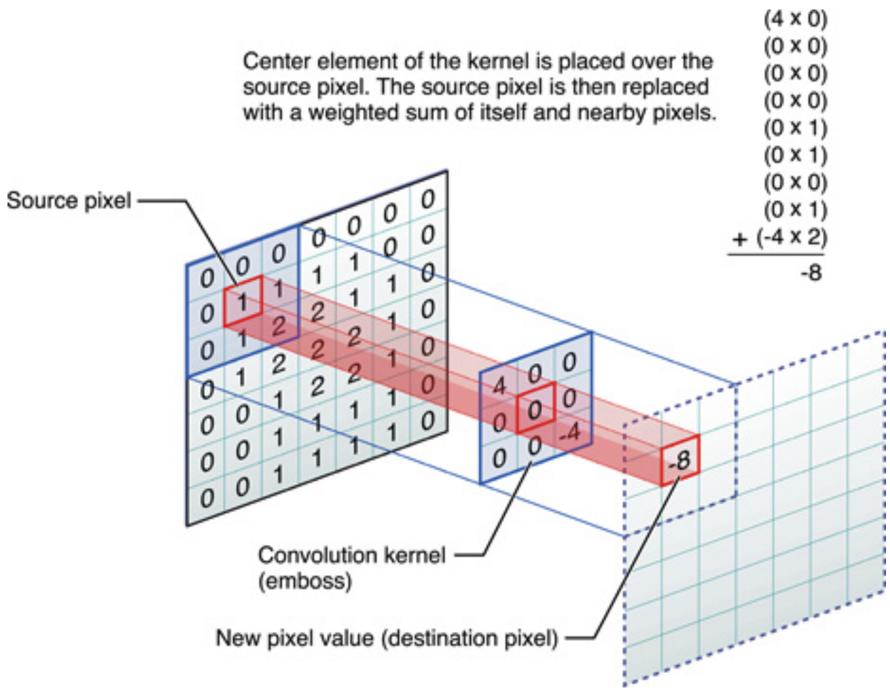
CNNs – key ideas



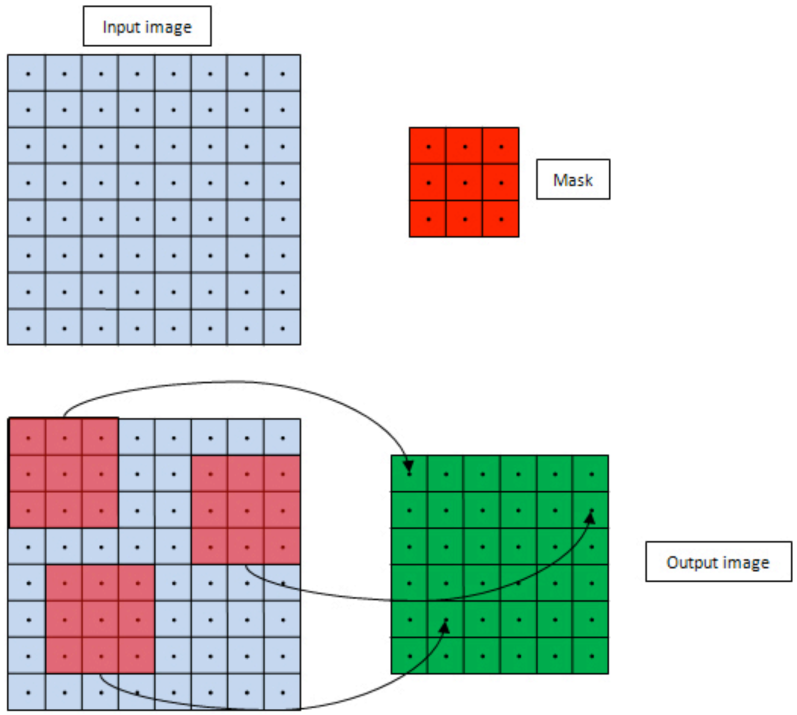
CNNs – key ideas



Convolution by Linear Filter

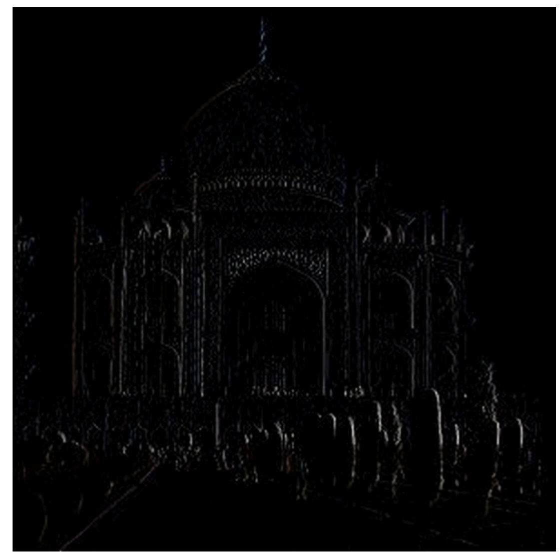


Convolution by Linear Filter

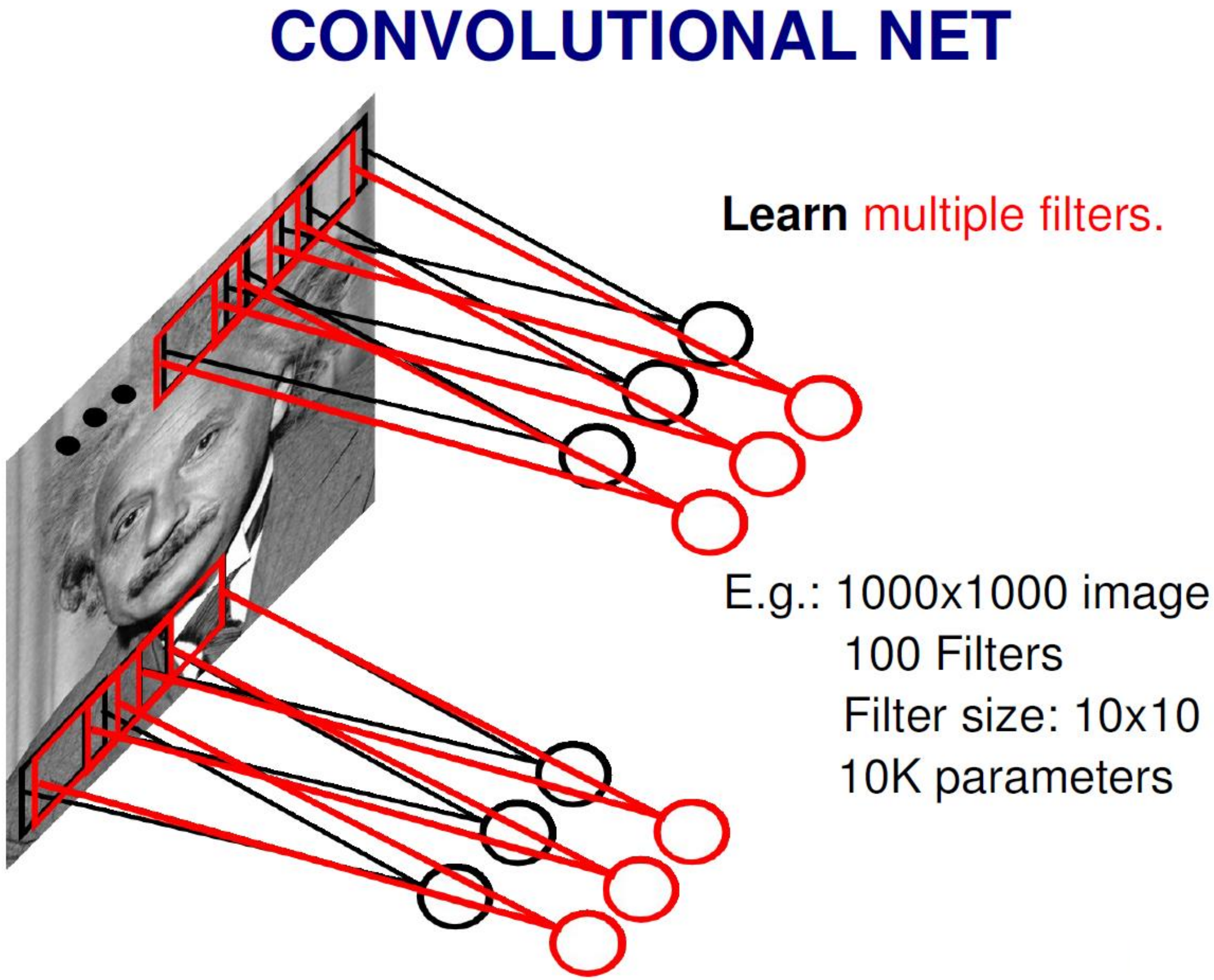


Example

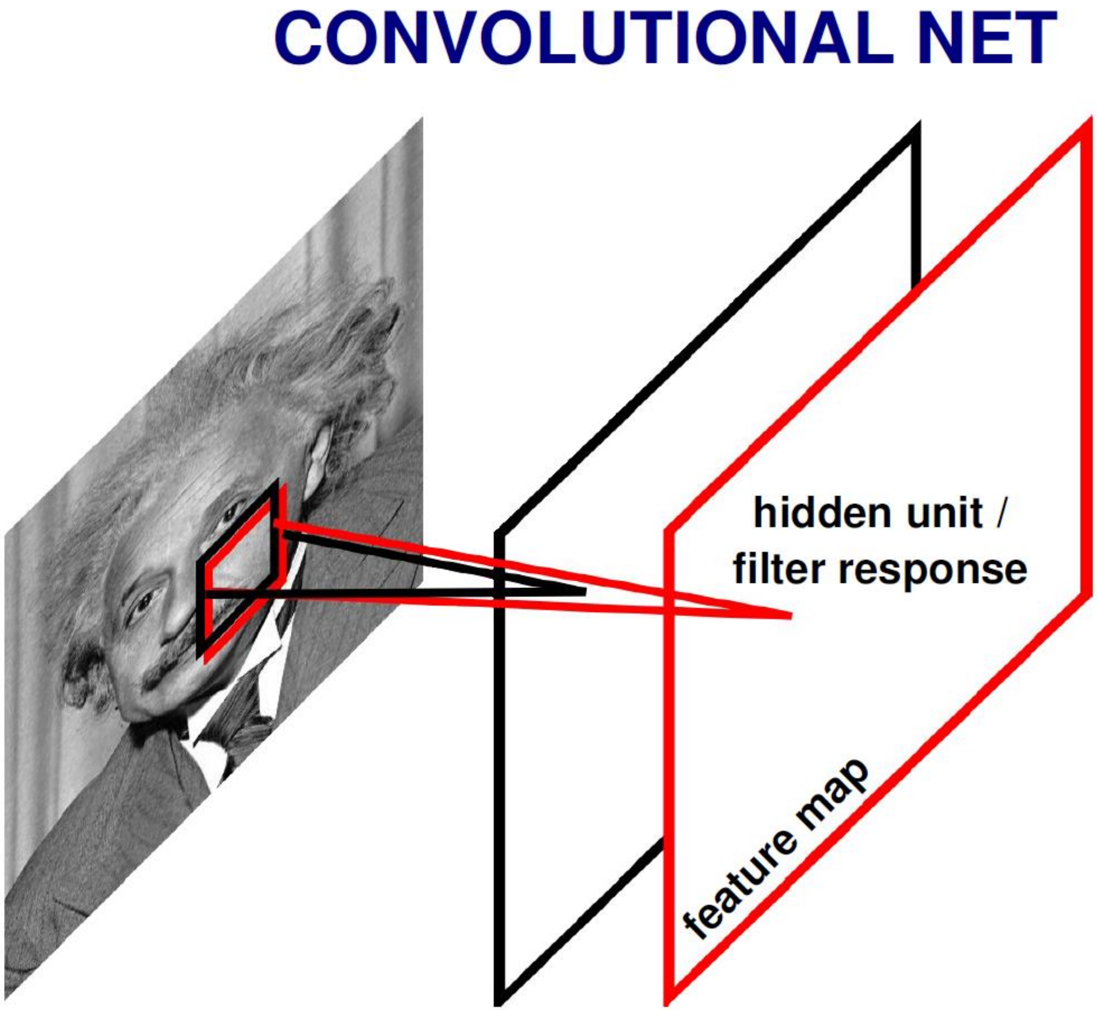
0	0	0	0	0
0	0	0	0	0
0	-1	1	0	0
0	0	0	0	0
0	0	0	0	0



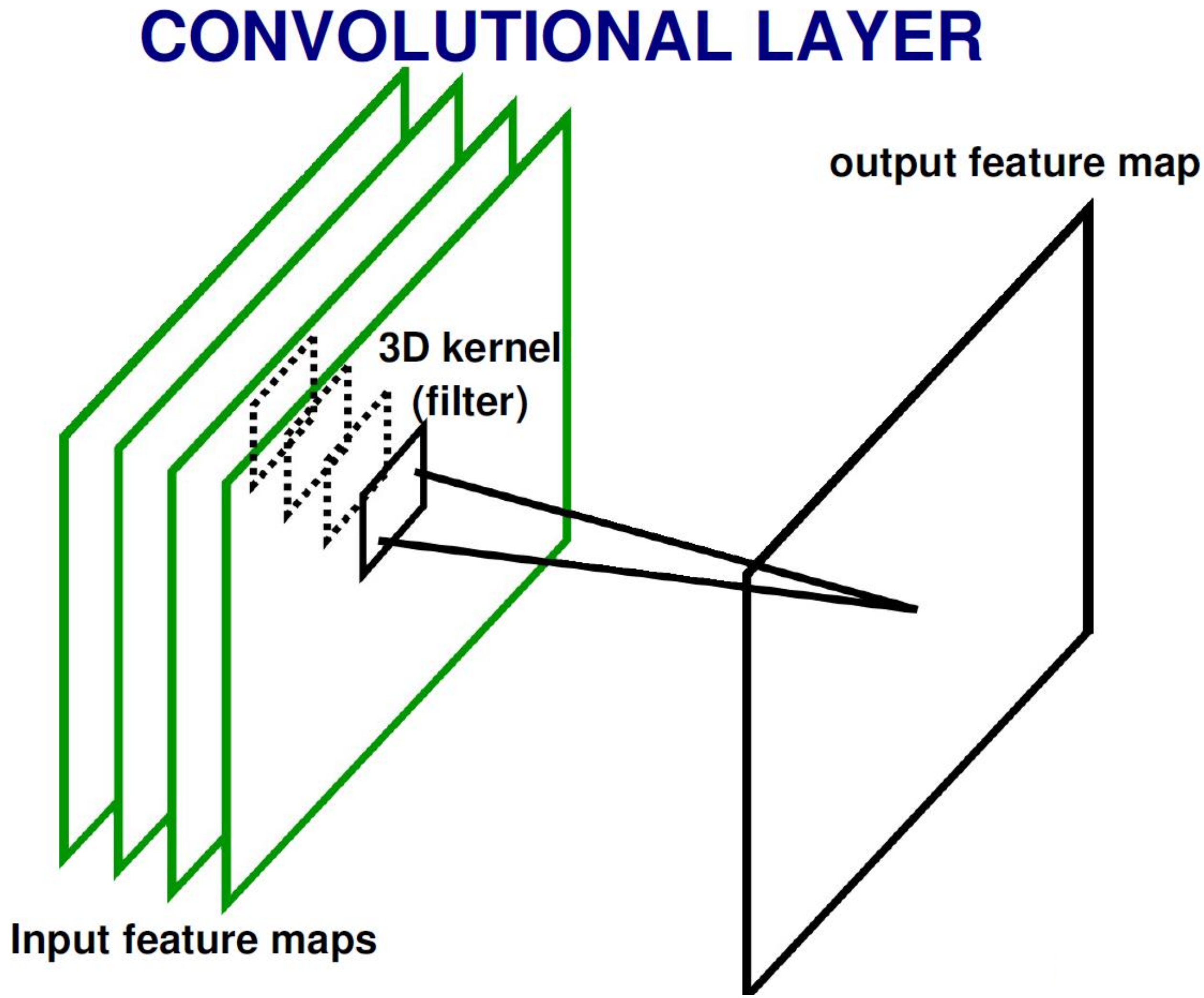
CNNs – key ideas



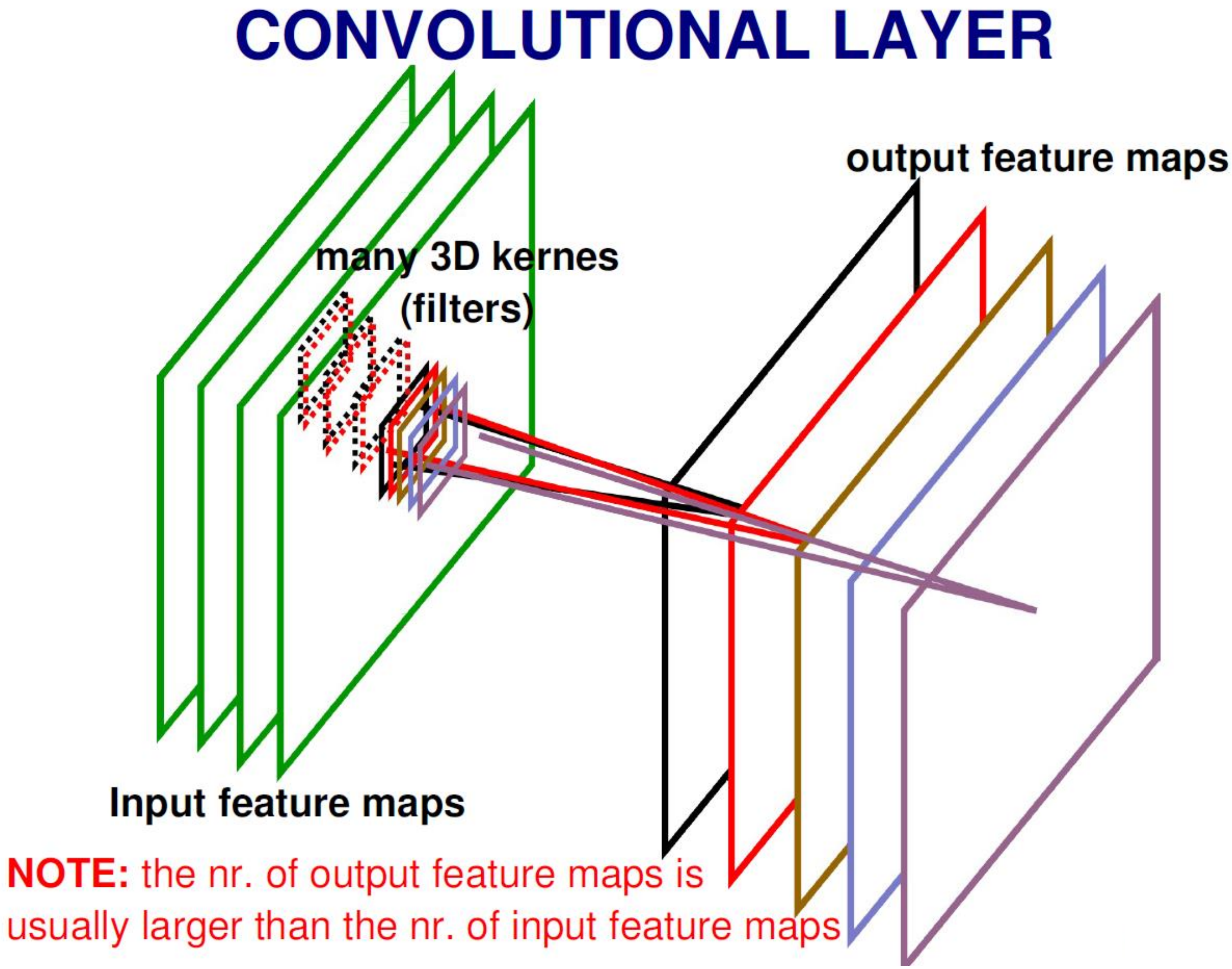
CNNs – key ideas



CNNs – key ideas

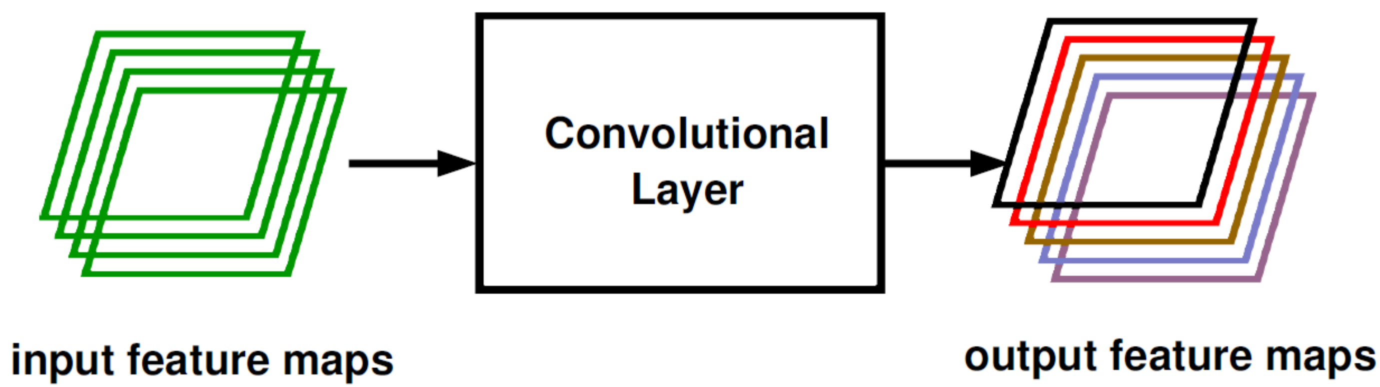


CNNs – key ideas



CNNs – key ideas

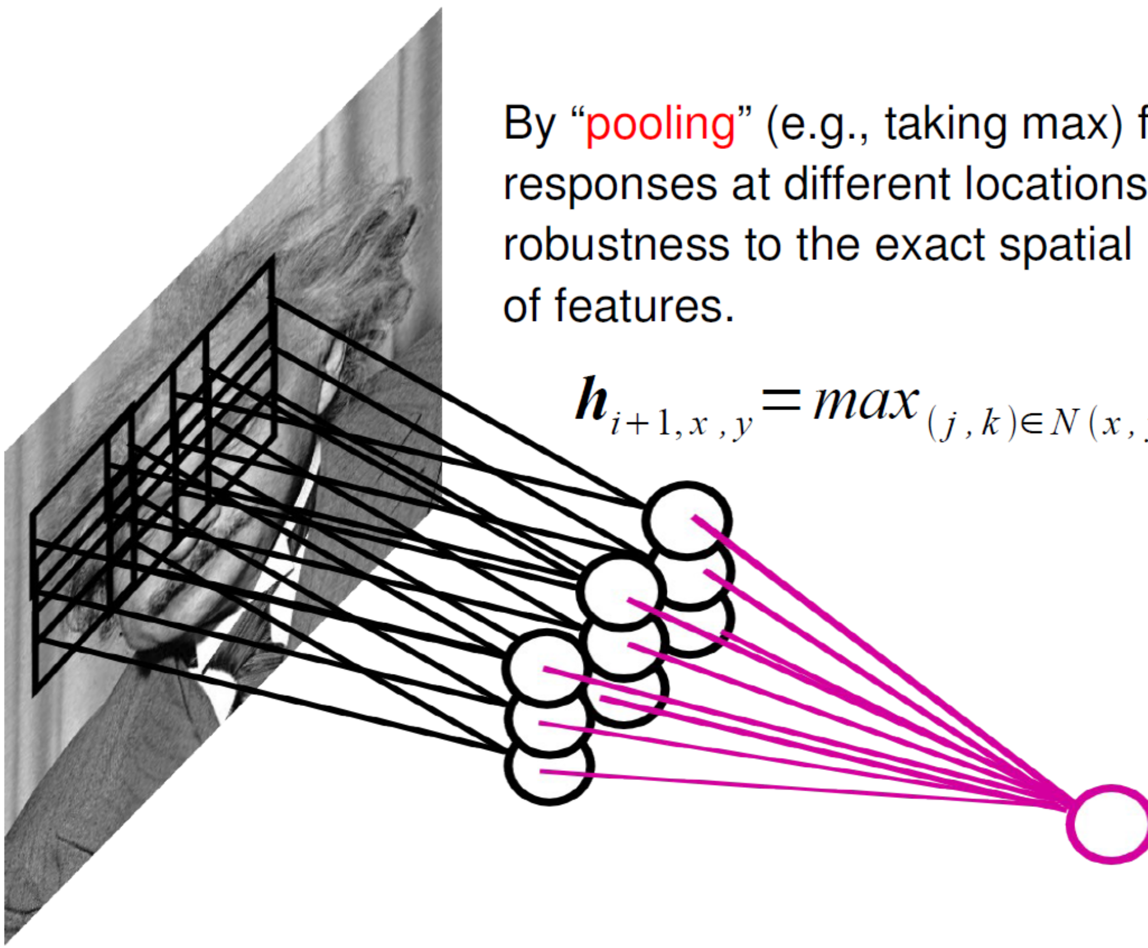
CONVOLUTIONAL LAYER



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

CNNs – key ideas

POOLING

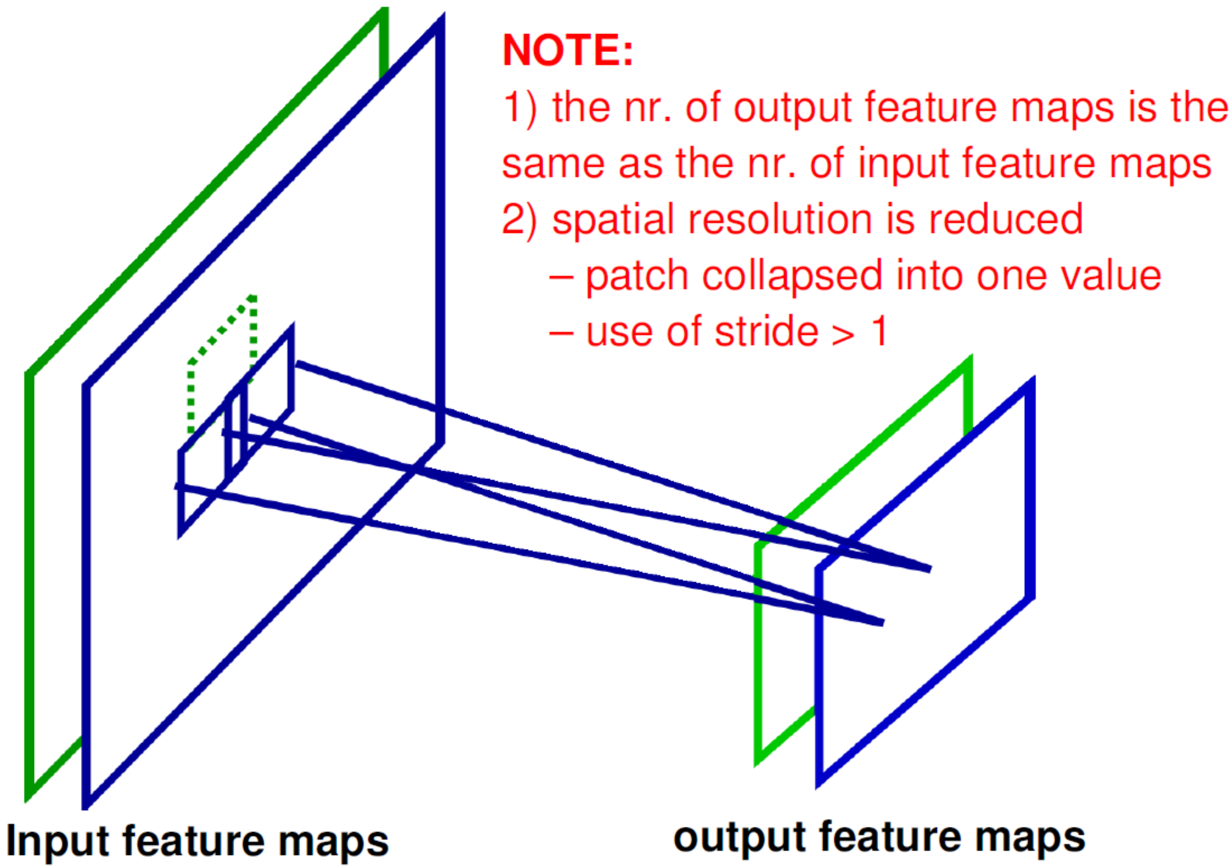


By "pooling" (e.g., taking max) filter responses at different locations we gain robustness to the exact spatial location of features.

$$h_{i+1,x,y} = max_{(j,k) \in N(x,y)} h_{i,j,k}$$

CNNs – key ideas

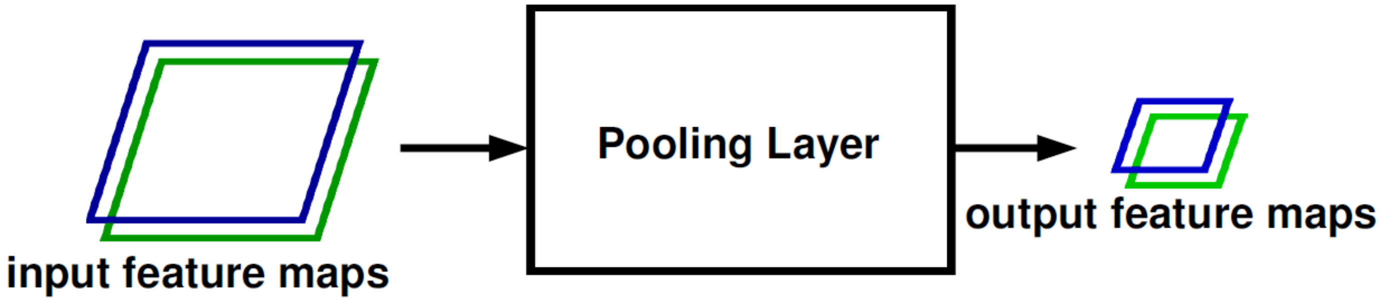
POOLING LAYER



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

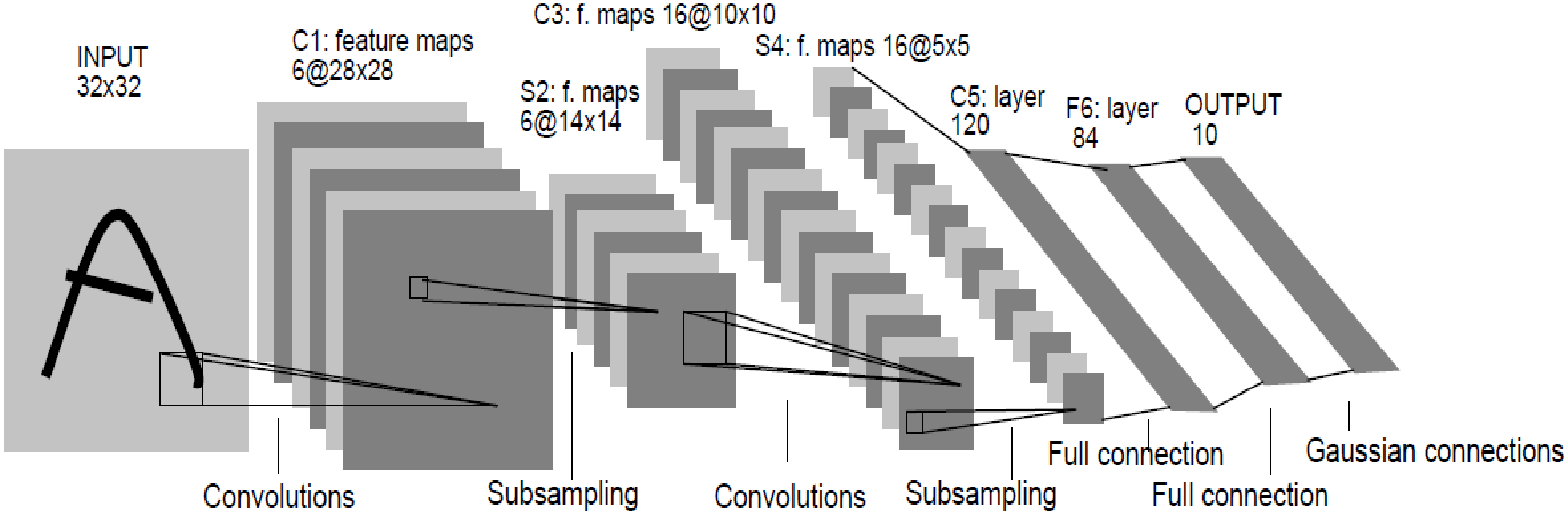
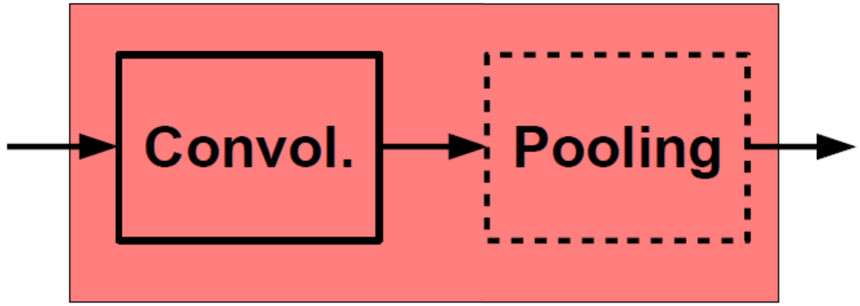


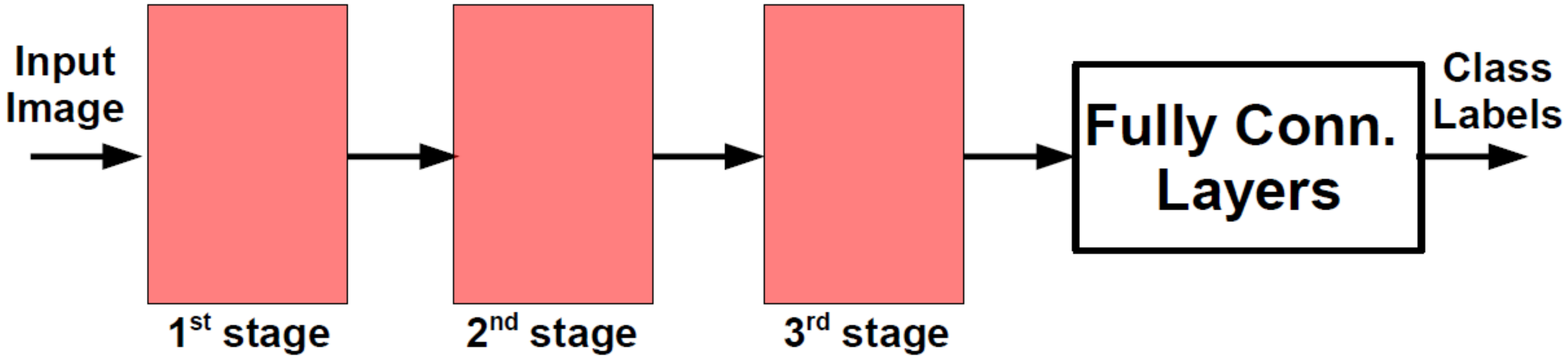
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

CNNs – typical architecture

One stage (zoom)

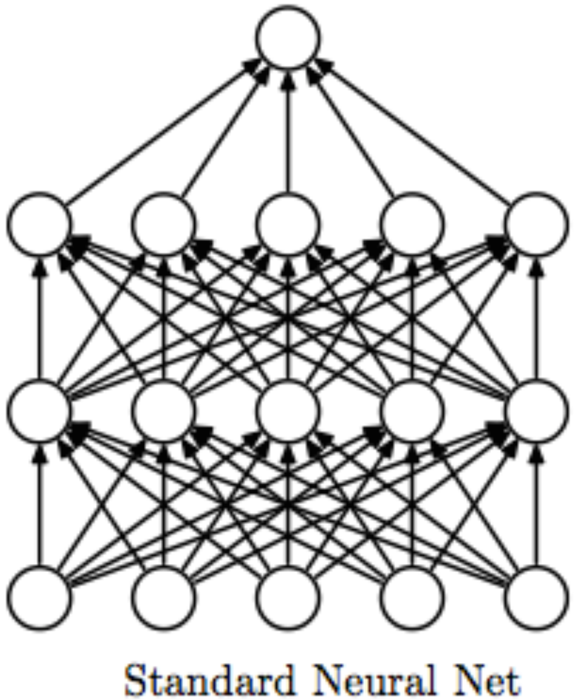


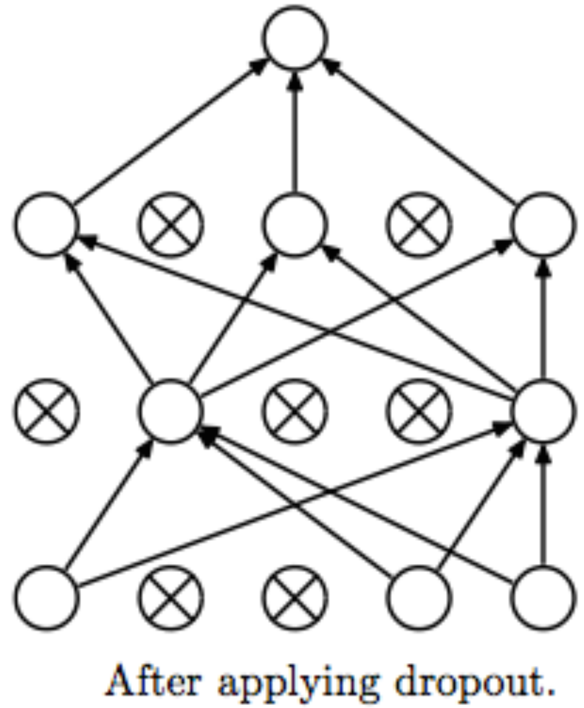
Whole system



Reducing Overfitting

Dropout





- With probability 0.5
- last two 4096 fully-connected layers.

Figure credit from [Srivastava et al.]

CNNs – conclusion

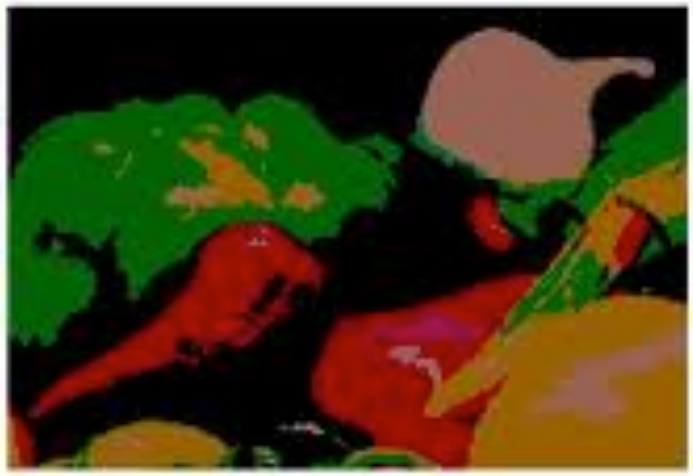
- Connect each hidden unit to a small patch of the input.
- Share the weight across hidden units.
- Subsampling layers are useful to reduce computational burden and increase invariance.

Successes

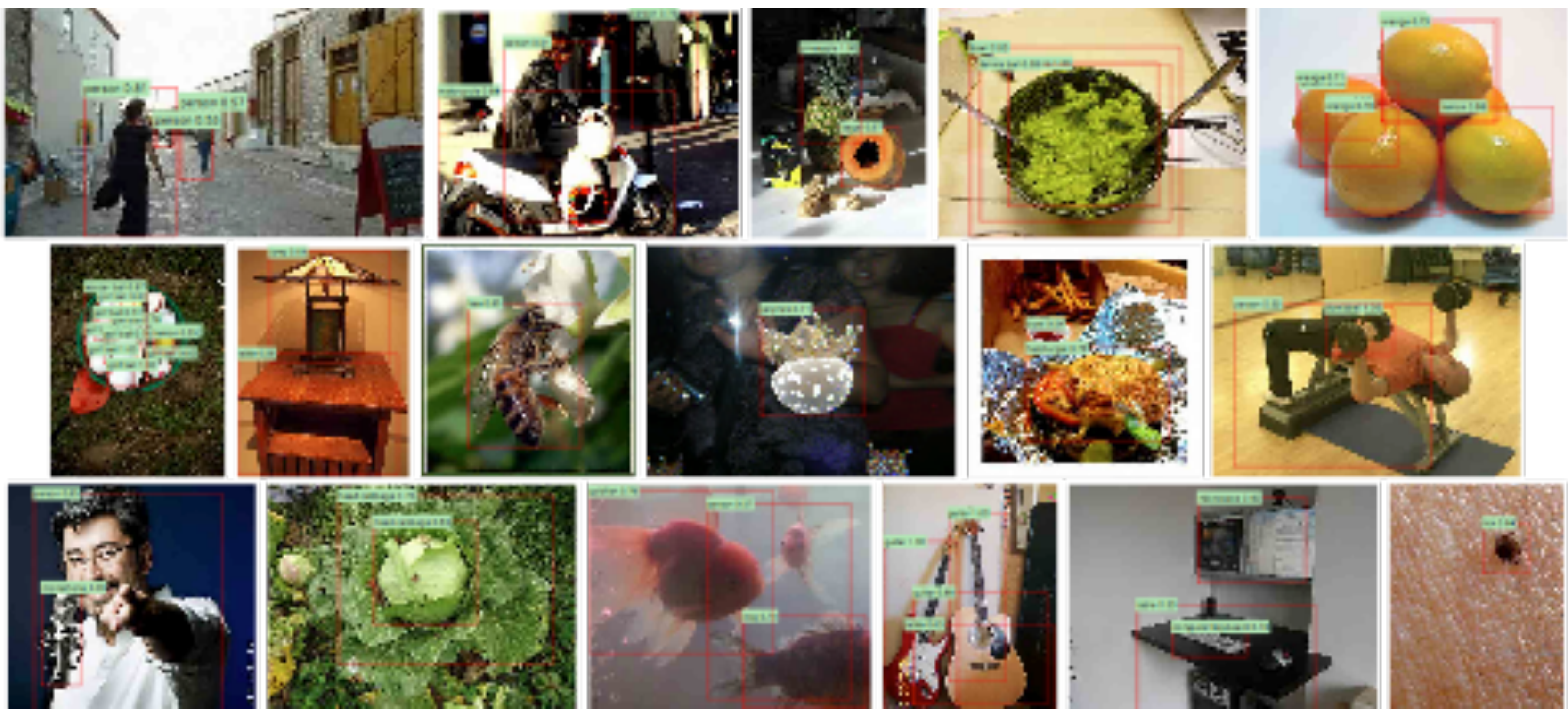
Computer vision
Speech
Chemistry

Object Classification

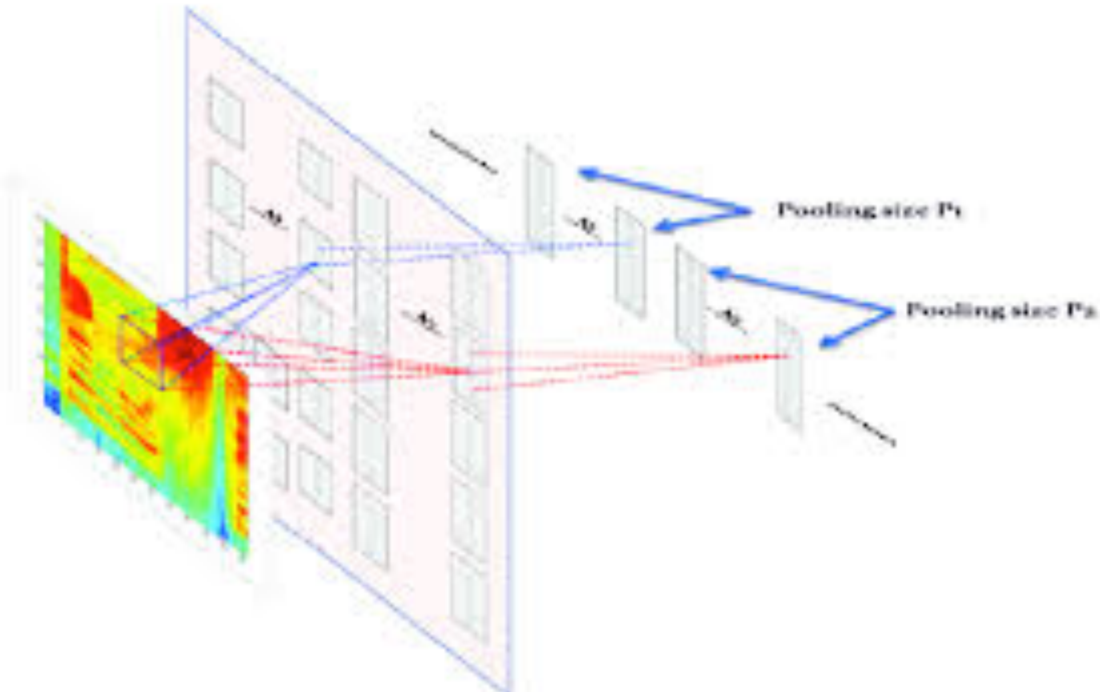
Segmentation



Object Detection



Speech



Physical Chemistry

Successfully predict atomization energy, polarizability, frontier orbital eigenvalues, ionization potential, electron affinity and excitation energies from molecular structure.