Convolutional Neural Networks

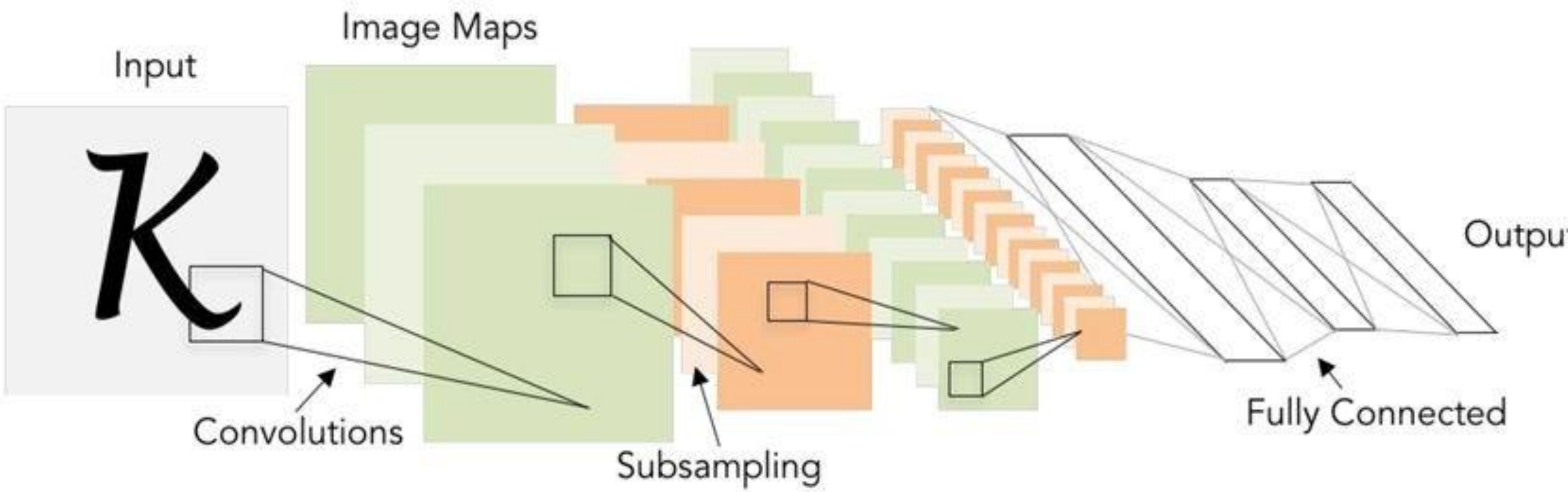
Artificial Intelligence

School of Computer Science
The University of Adelaide

You can think of a deep network as a multistage information-distillation operation, where information goes through successive filters and comes out increasingly purified. (François Chollet, Deep Learning with Python (Shelter Island, NY: Manning Publications, 2018)

Convolutional Neural Networks (CNN)

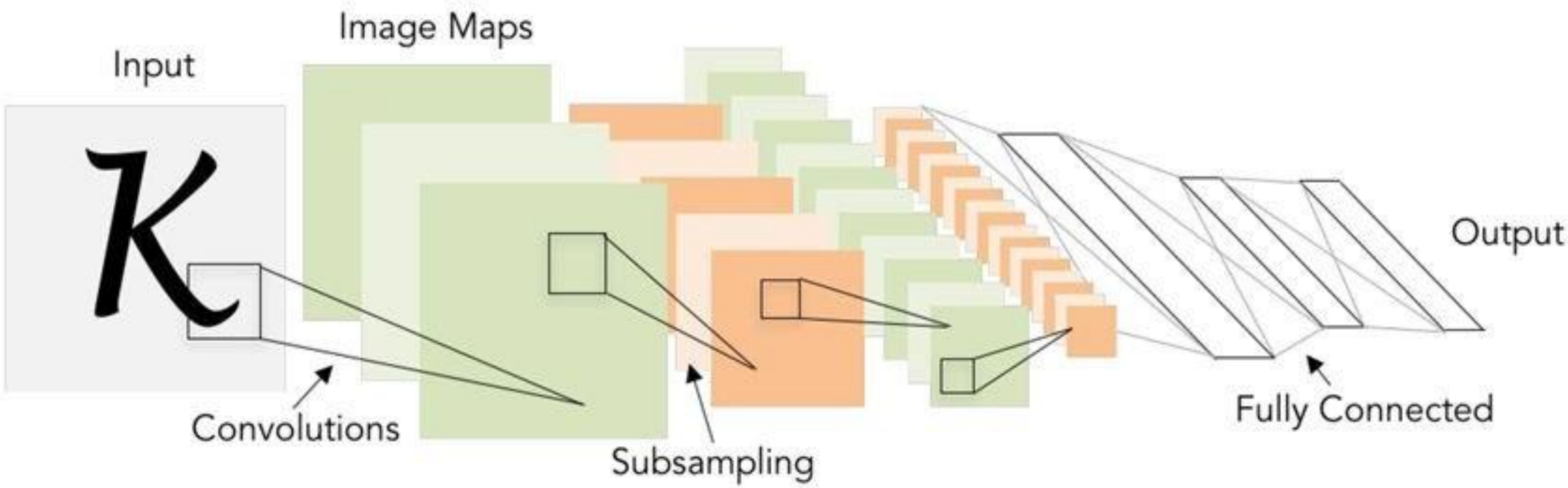
Introduced by Lecun et al. 1989.



LeNet-5

- Addressed the problem overfitting due to the explosion of the number of parameters as the networks become deep.
- Convolutional NN : Low number of parameters in deeper net.
- Works very well on images.

Building Blocks of Deep CNNs

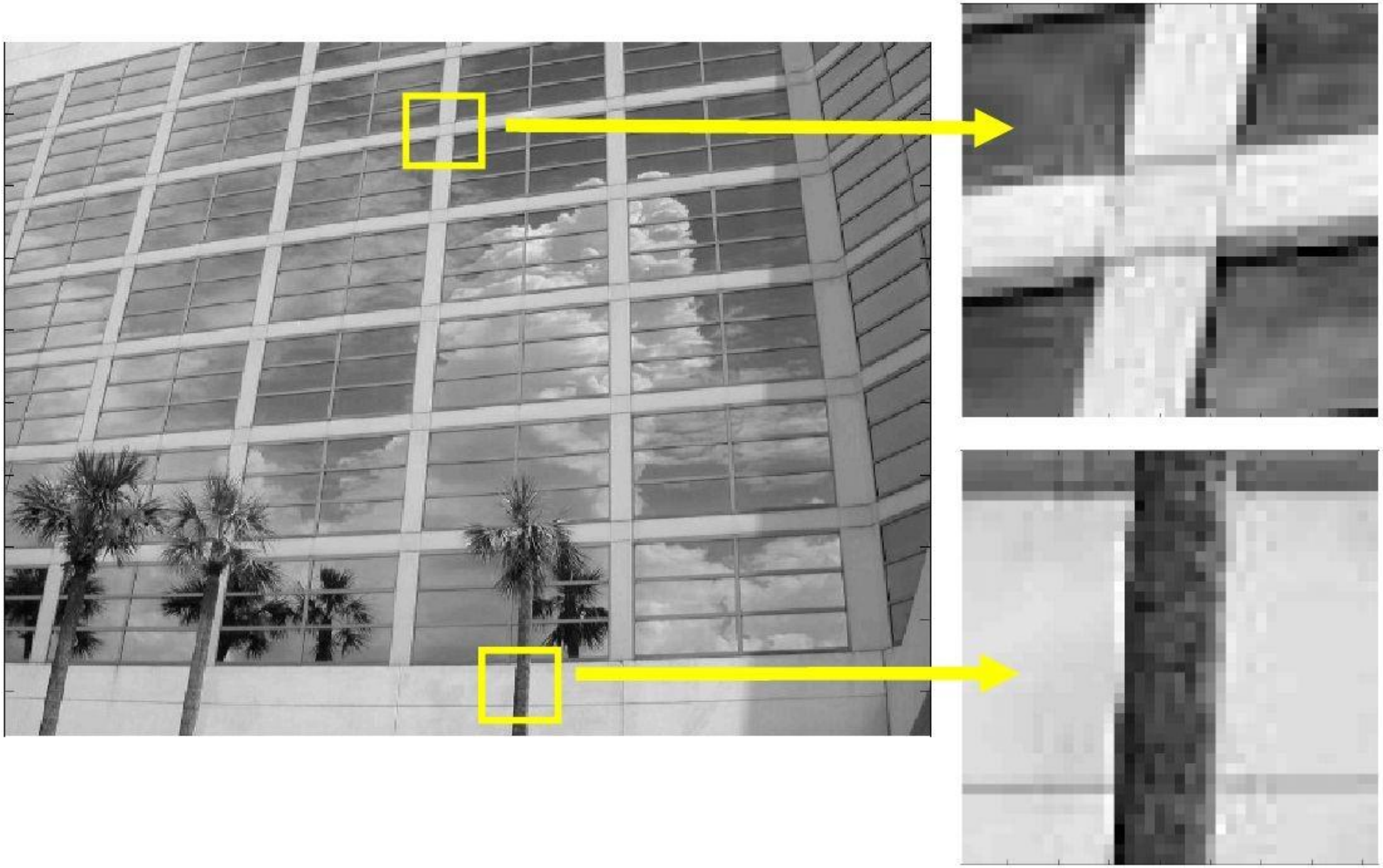


LeNet-5

- Convolution layers - replaces many fully connected layers.
- Subsampling layers - max pooling, average pooling…
- Fully connected layers
- Activations - mostly Rectified Linear Units (ReLu) these days.

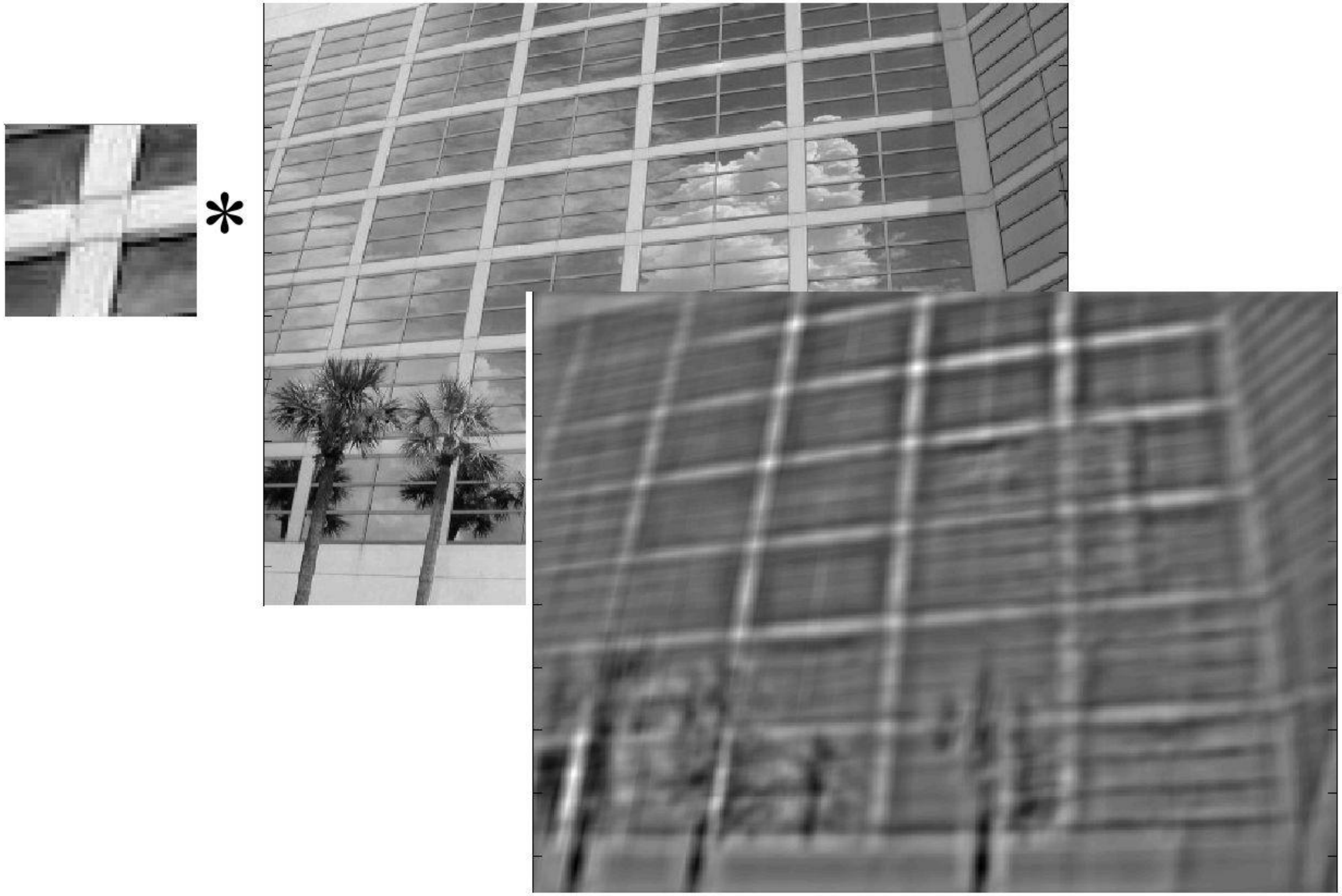
Convolution - Simple Pattern Detector

Convolving a filter with and image = detecting a template.



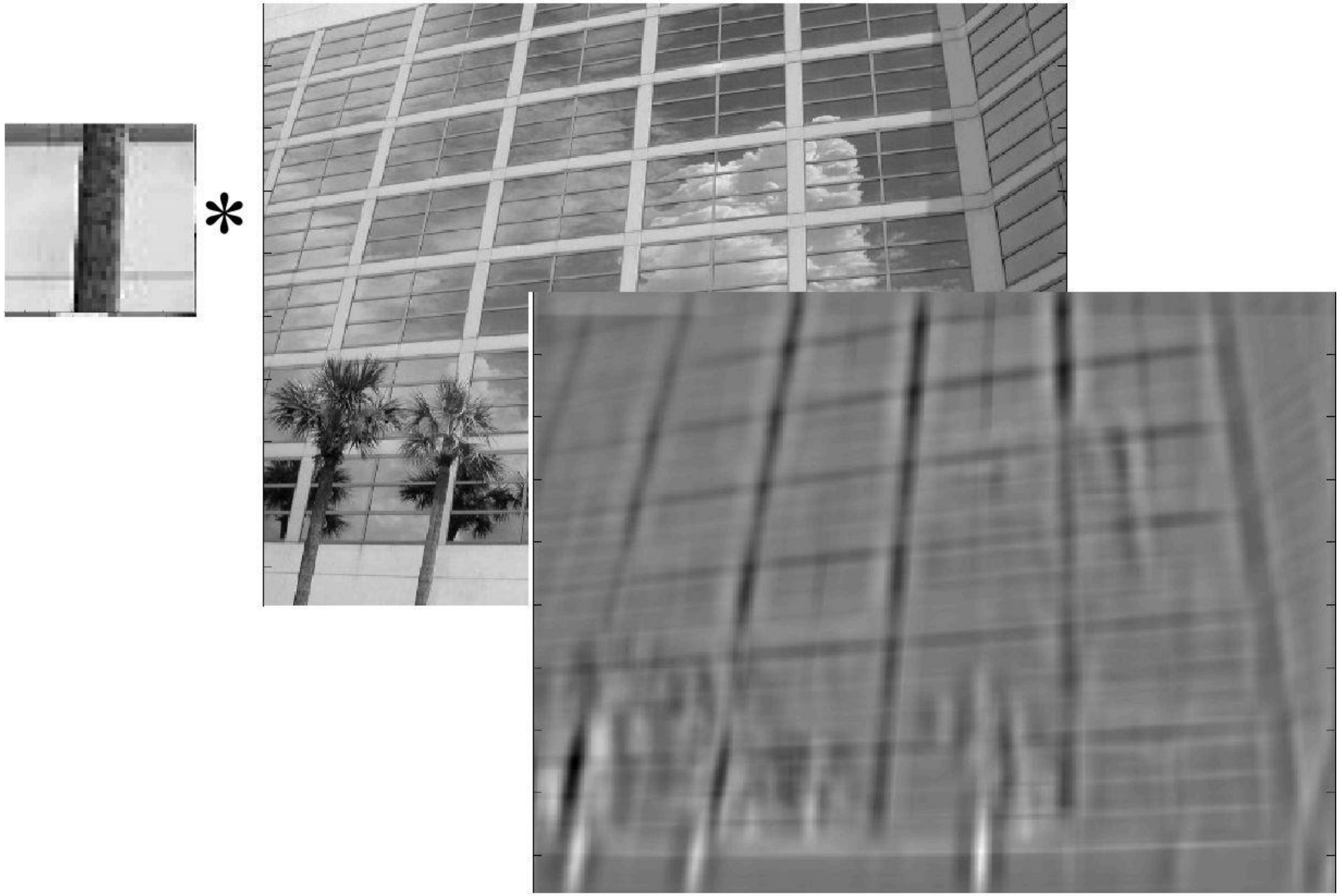
Convolution - Simple Pattern Detector

Give maximum response where a local image region best match a template.



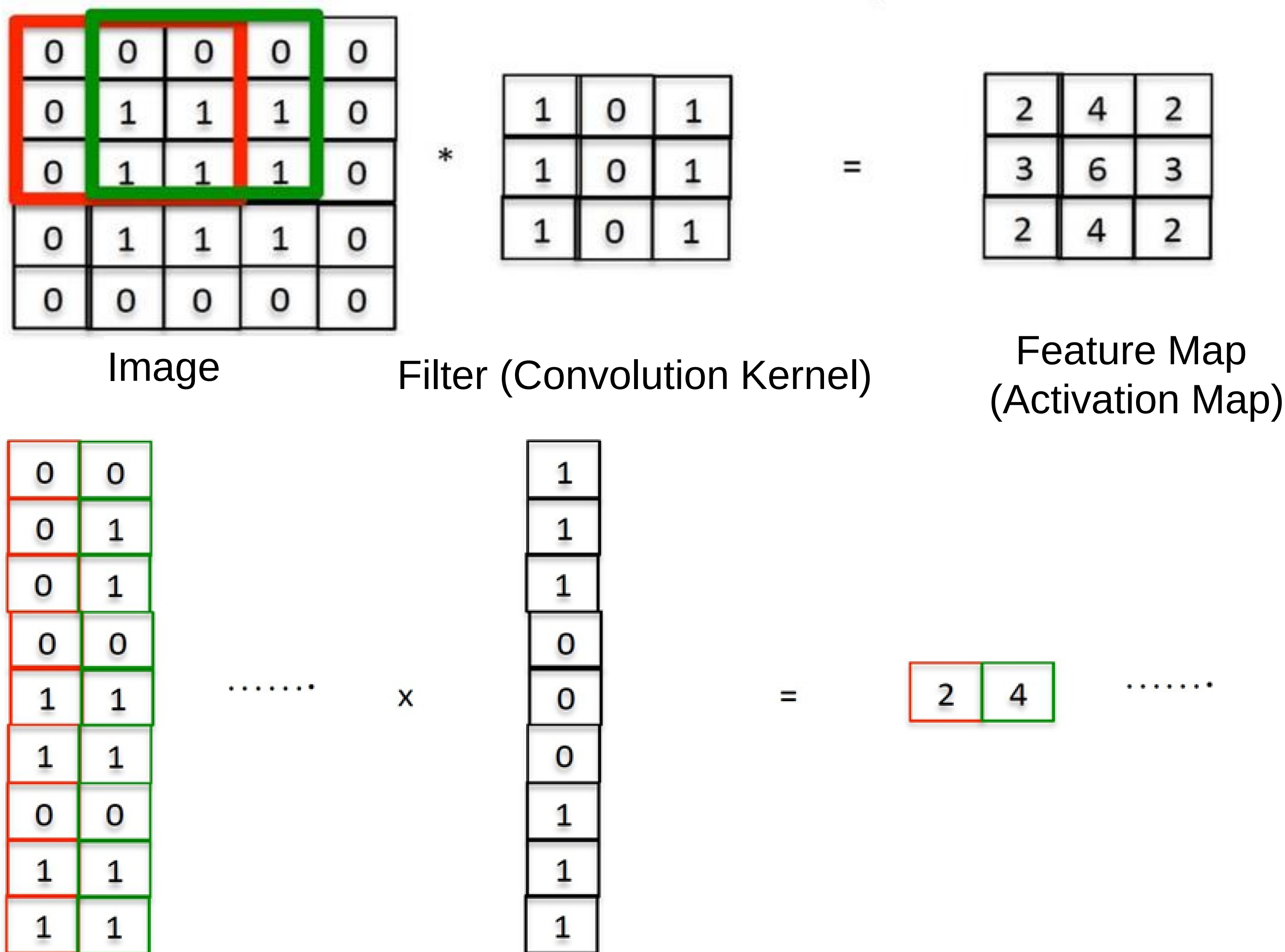
Convolution - Simple Pattern Detector

You can match multiple templates.

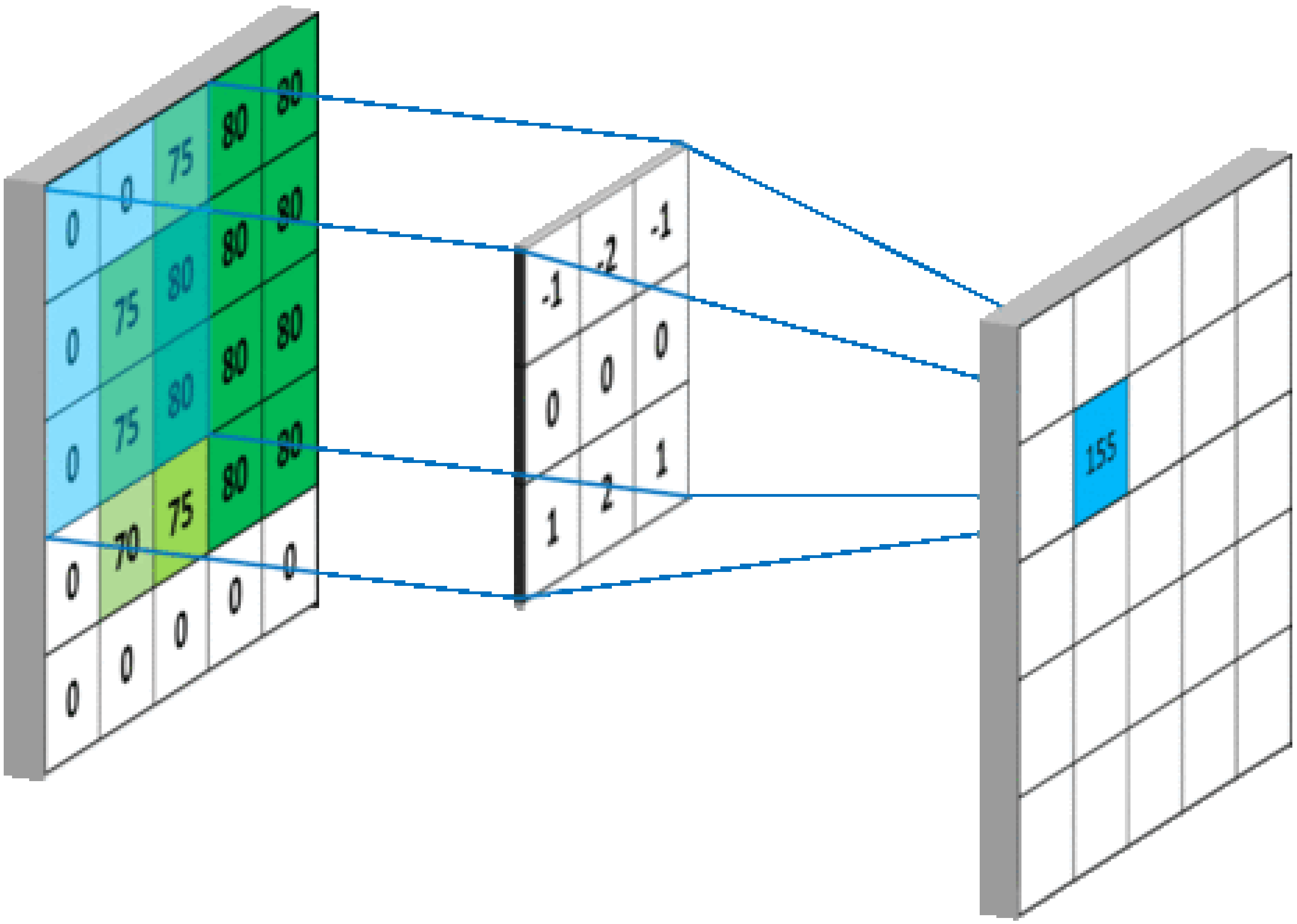


Robert Collins, CSE486, Penn State

How is convolution done in practice?



Convolution Operator (2D)



Convolution Operator (2D)

$$o[i,j] = \sum_m \sum_n f[i-m, j-n] * g[m,n]$$

$$\begin{bmatrix} \boxed{\begin{matrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{matrix}} & \begin{matrix} \cdots & a_{0,n} \\ \cdots & a_{1,n} \\ \cdots & a_{2,n} \end{matrix} \\ \begin{matrix} \cdots & \cdots & \cdots \\ a_{m,0} & a_{m,1} & a_{m,2} \end{matrix} & \begin{matrix} \cdots & \cdots \\ \cdots & a_{m,n} \end{matrix} \end{bmatrix}$$

Image

$$f = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \qquad g = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix}$$

Portion of Image

Filter (Cov. kernel)

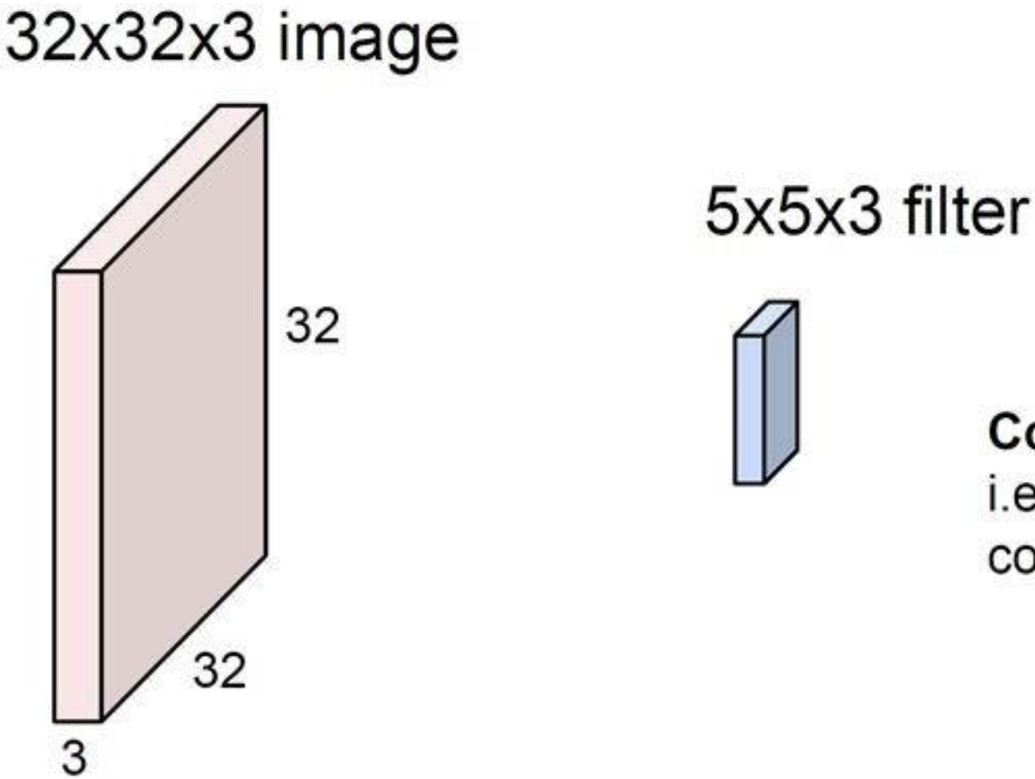
Convolution Operator (2D)

$$o[i,j] = \sum_m \sum_n f[i-m, j-n] * g[m,n]$$

$$f = \begin{bmatrix} \boxed{a_{0,0}} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix} \qquad g = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & \boxed{b_{1,1}} \end{bmatrix}$$

$$c_{1,1} = a_{0,0} b_{1,1}$$

Convolutional Layer



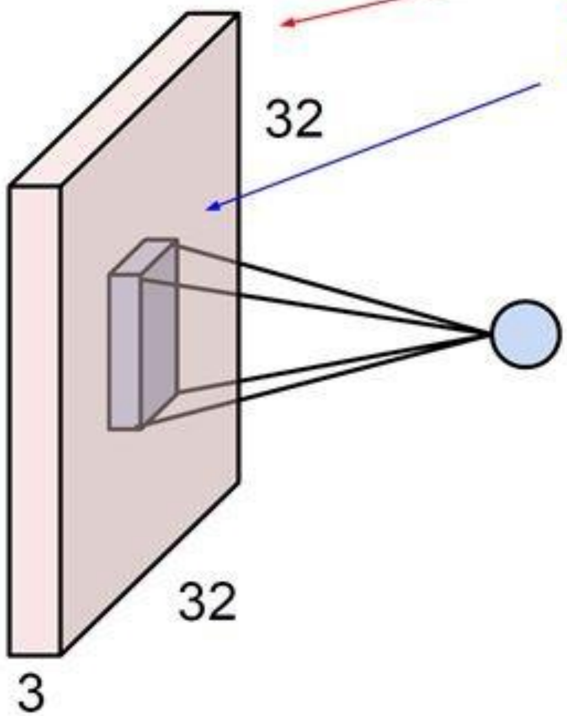
Convolve the filter with the image
i.e. "slide over the image spatially,
computing dot products"

RGB: red, green, blue
000: black
111: white

Convolutional Layer

32x32x3 image

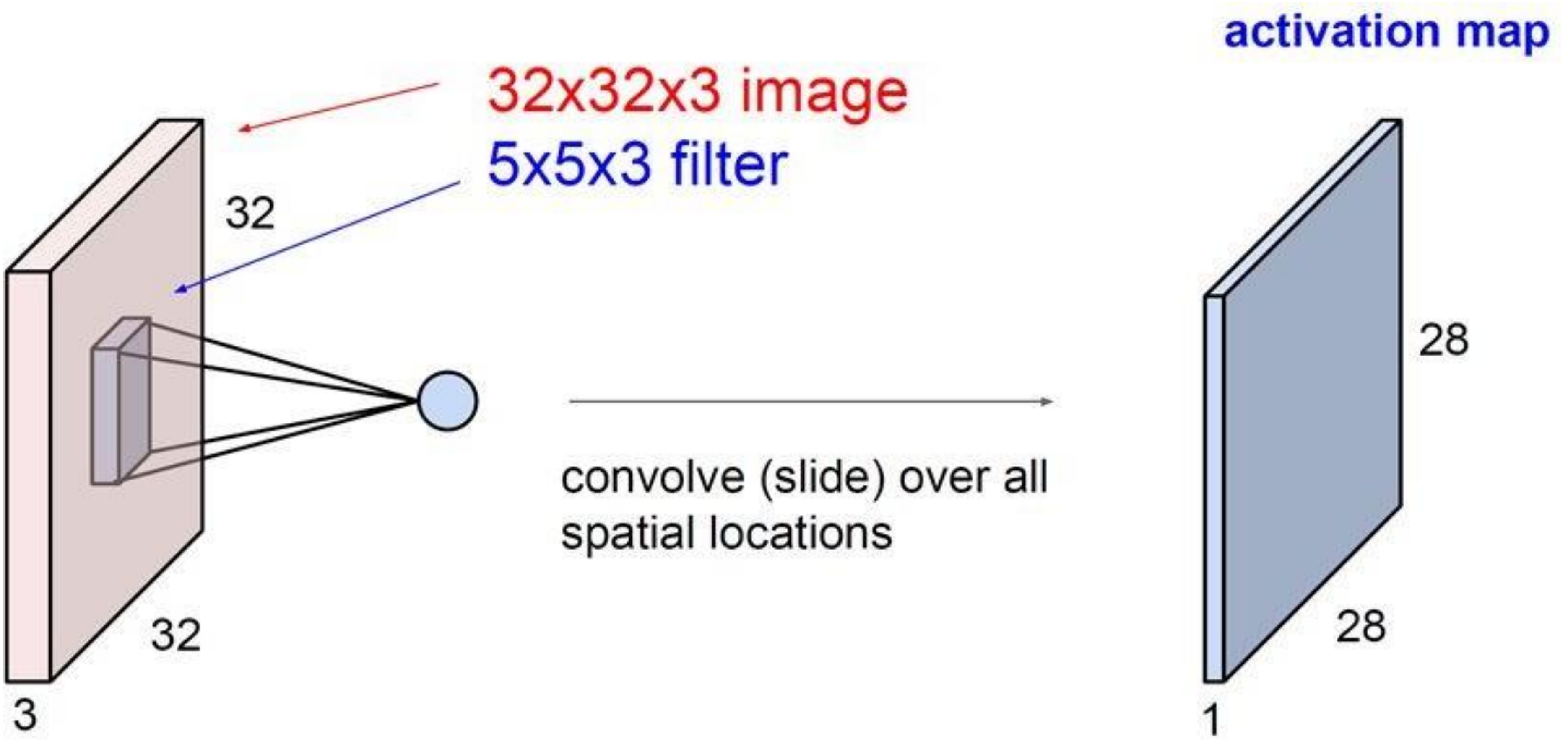
5x5x3 filter w



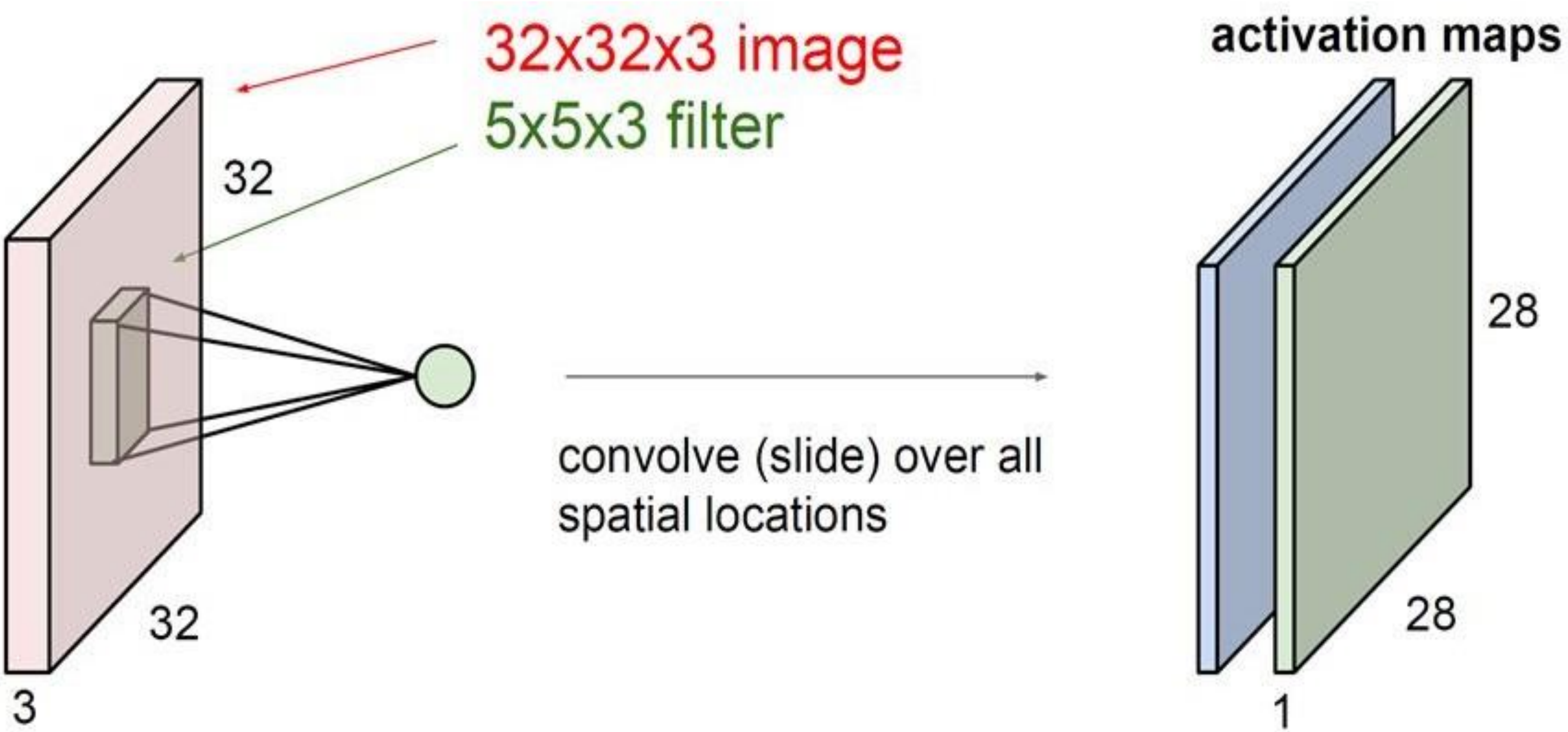
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

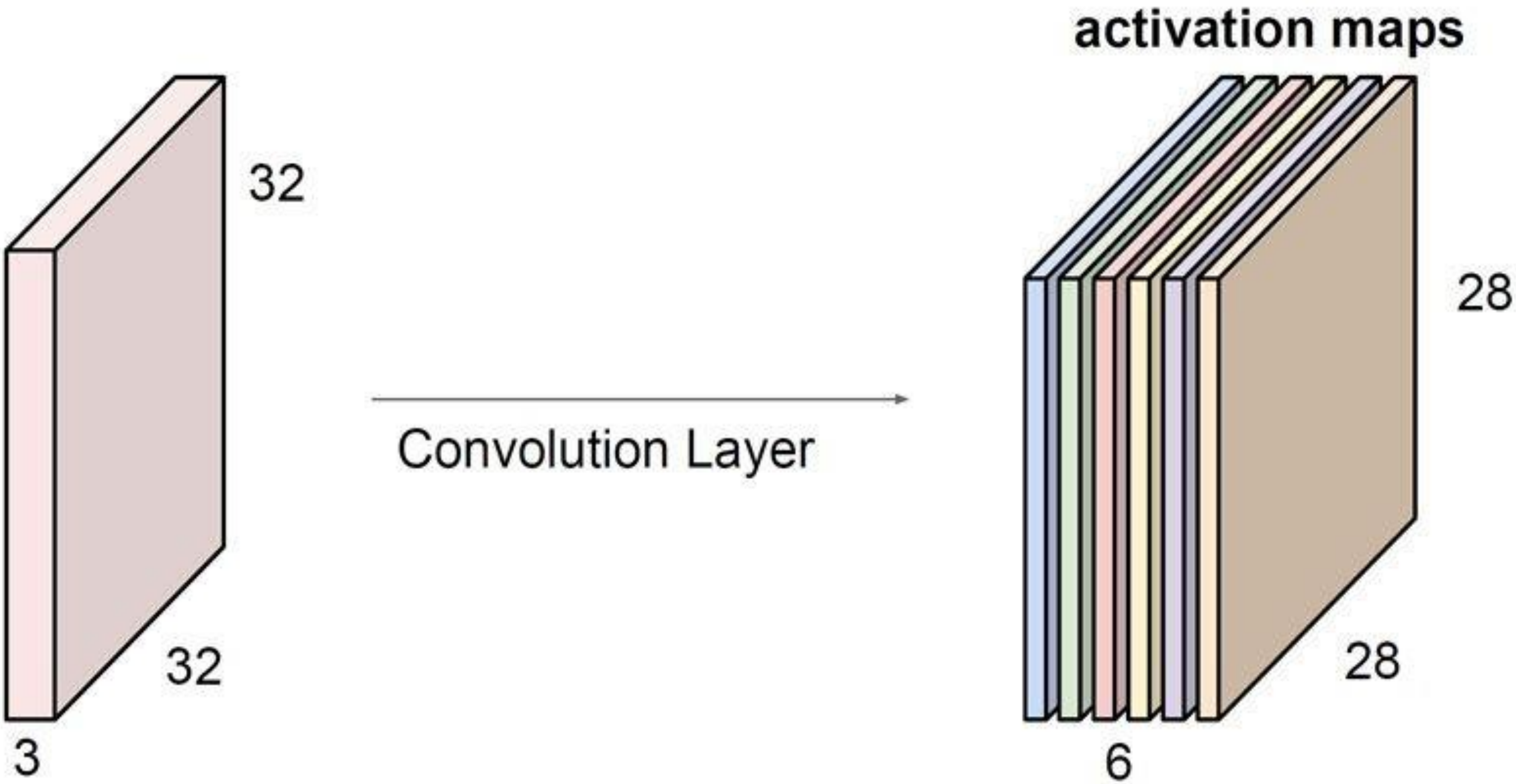
Convolutional Layer



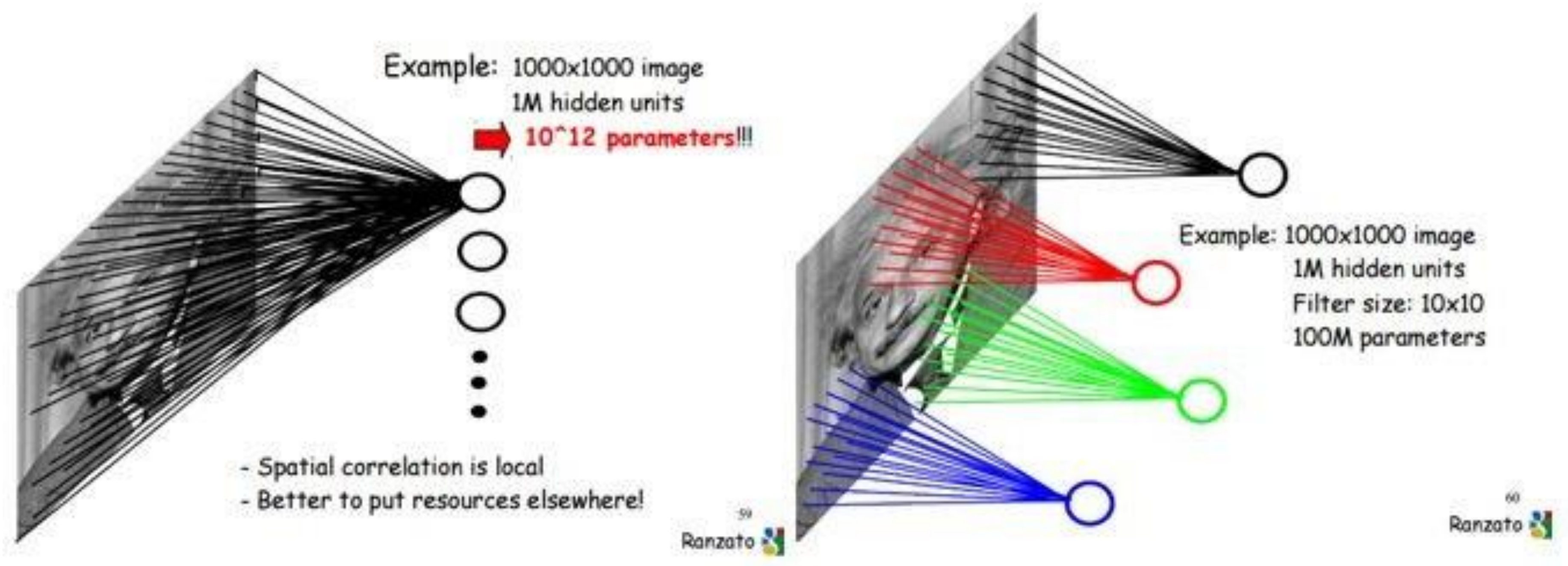
Convolutions More Filters



Convolution Layer

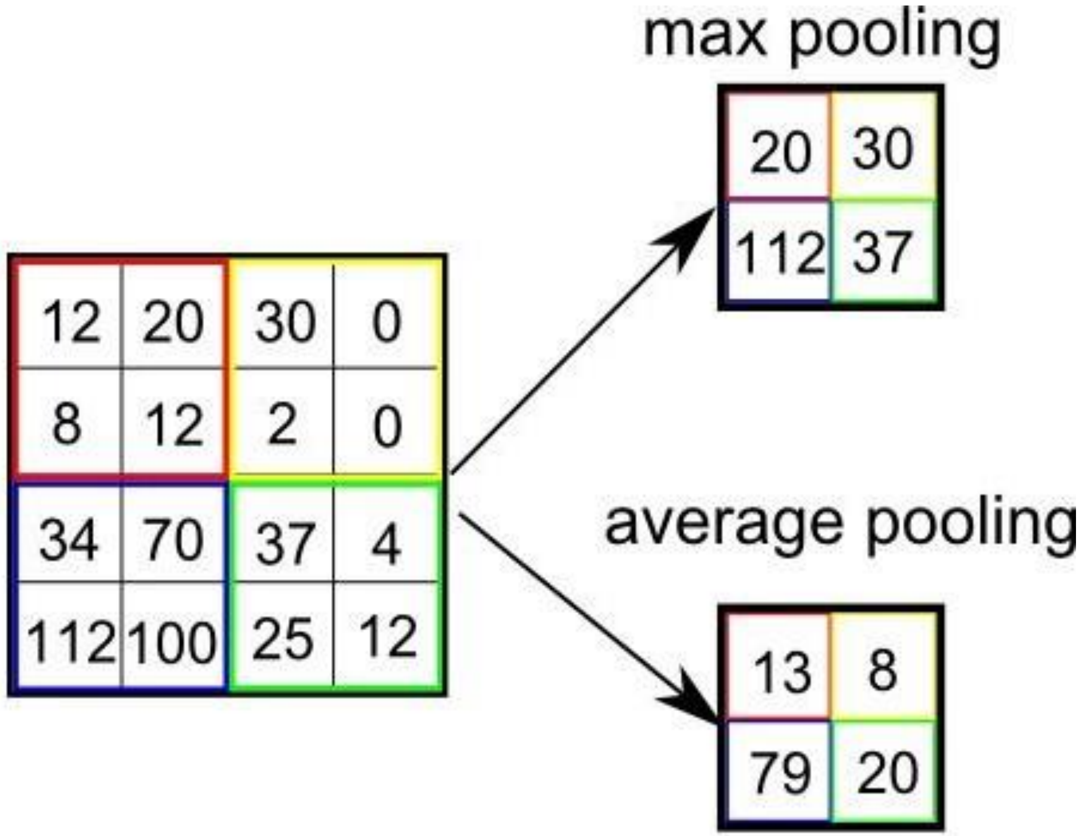


Why Convolutions?



- Every output neuron has sparse connectivity - more tractable.
- **Weight Sharing** - detects repeated local structures in the data.
  - 1000 x 1000 image,
  - MLP: 1M hidden units (MLP): 10^12 parameters.
  - CNN: 1M filters with size 10x10 (100 weights each)
    100M parameters

Subsampling - Pooling



- Max Pooling
- Average Pooling

Subsampling - Pooling

Activation Map (Feature Map)

2	0	-2	0
3	0	-3	0
2	0	-2	0
0	0	0	0

Max pooling

3	0
2	0

Activation Map (Feature Map)

0	2	0	-2
0	3	0	-3
0	2	0	-2
0	0	0	0

Max pooling

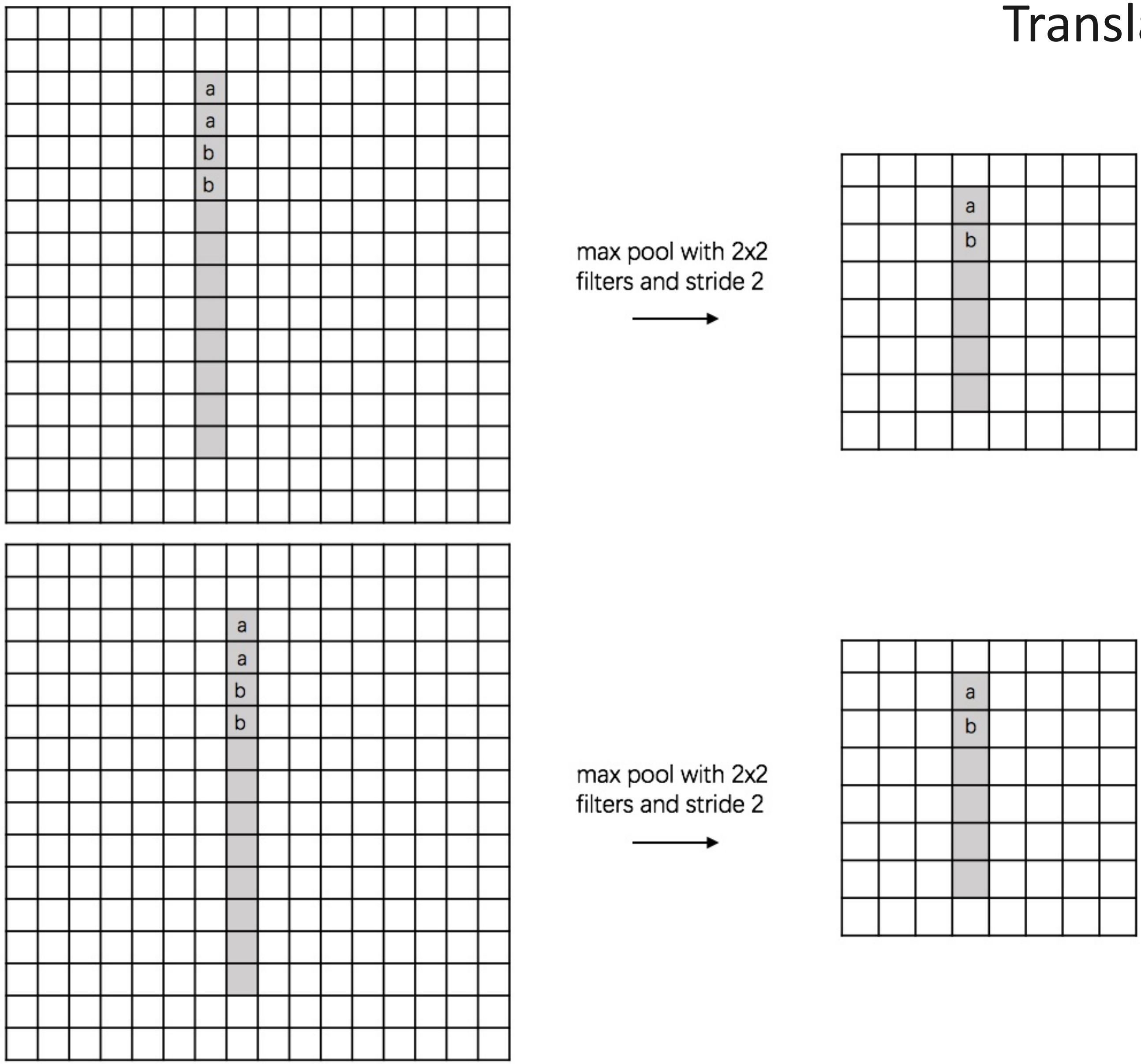
3	0
2	0

- Reduces size of the data and thus computation cost.
- Add translation invariance - Small horizontal or vertical translations does not affect the outputs.

Translation Invariance

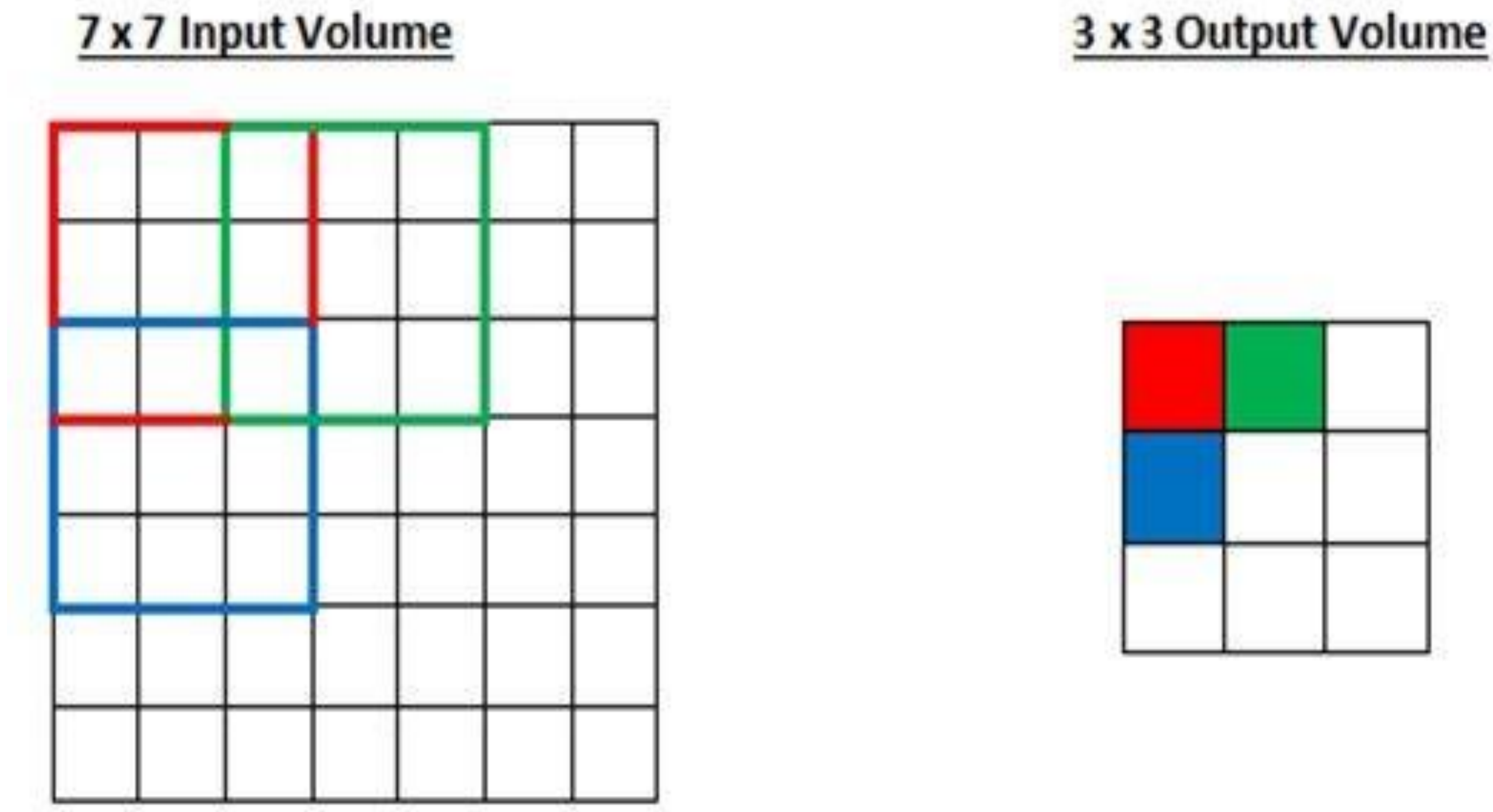
Subsampling - Pooling

Translation invariance

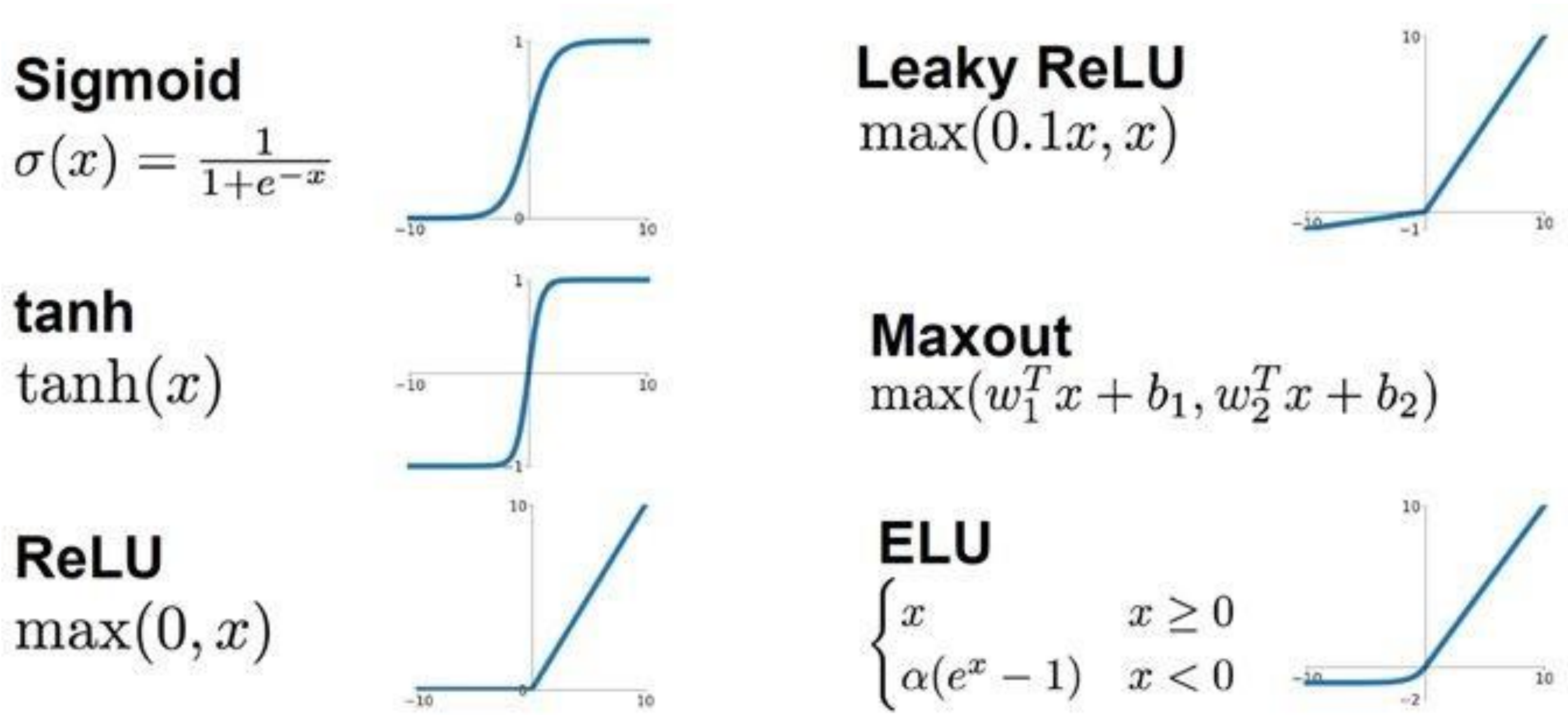


Convolutions with Strides

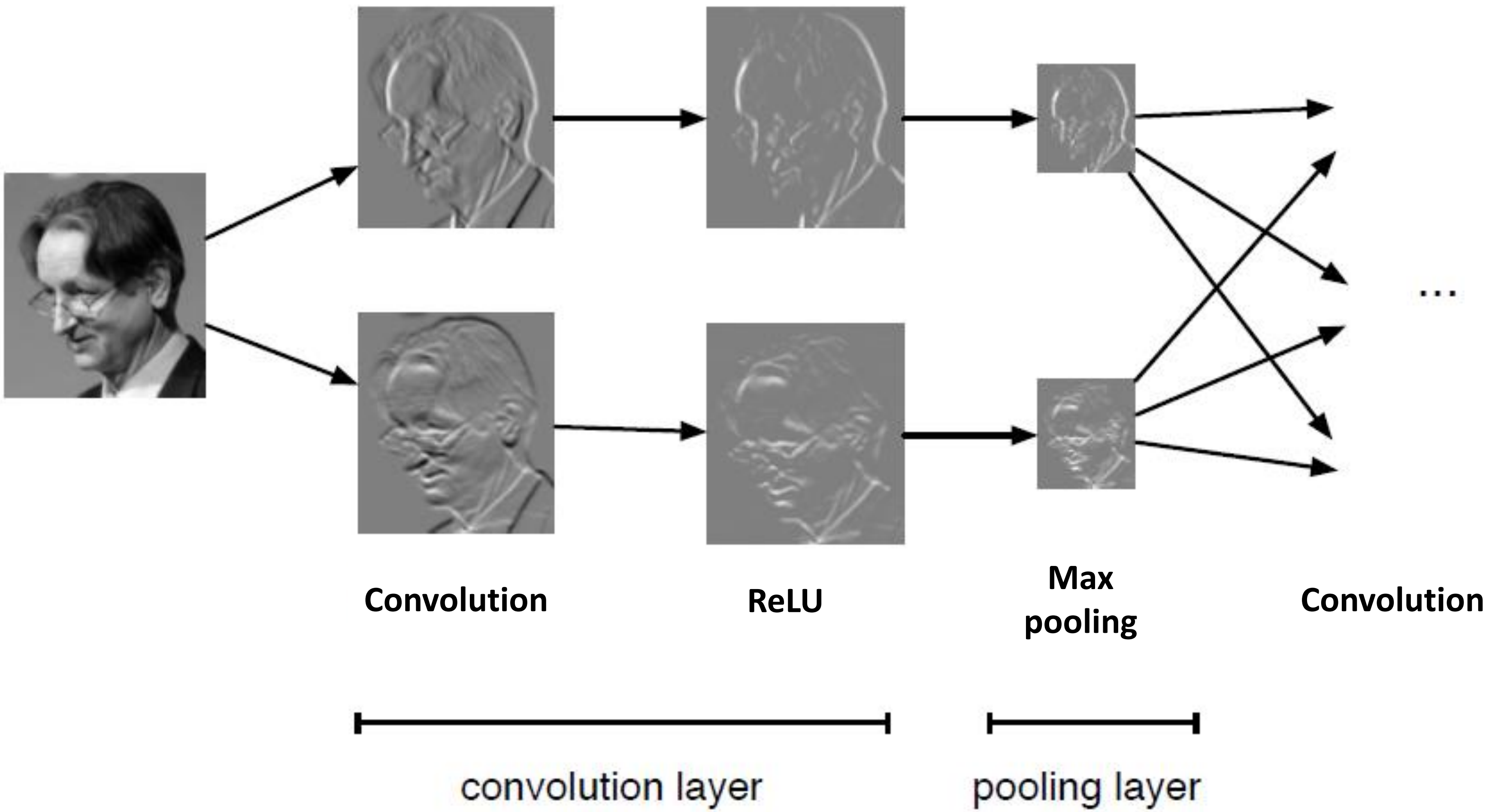
- Also reduces the size of the output.
- Can be alternative to pooling for subsampling.



Remember Nonlinear Activations?

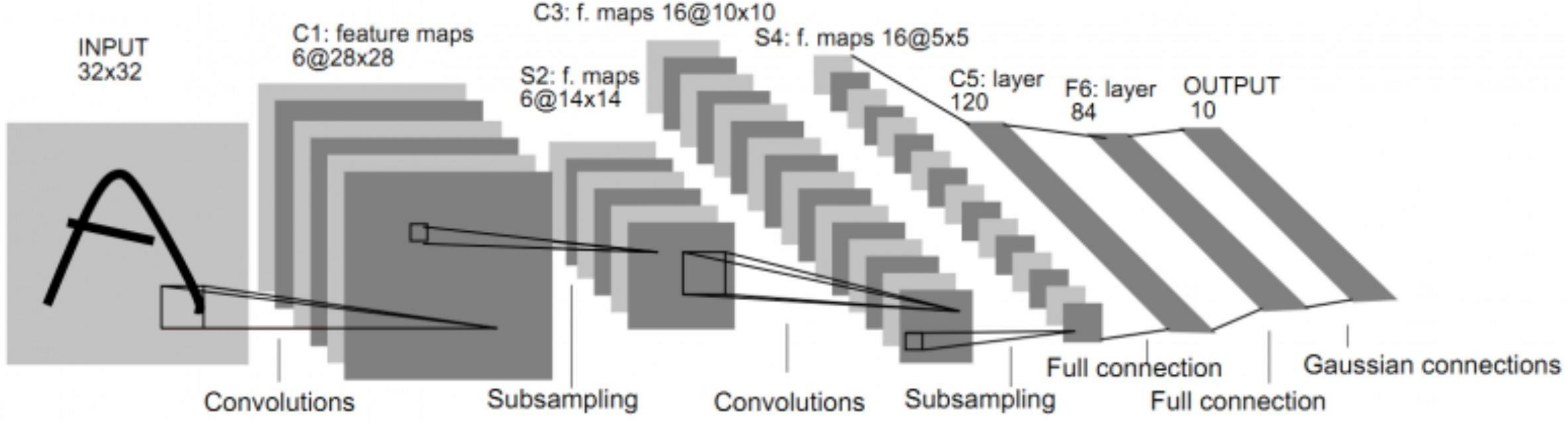


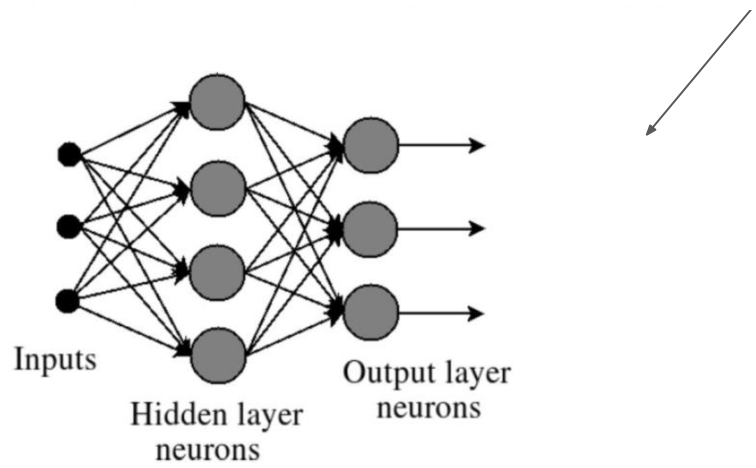
Activations in CNN



Roger Grosse, CSC321, University of Toronto

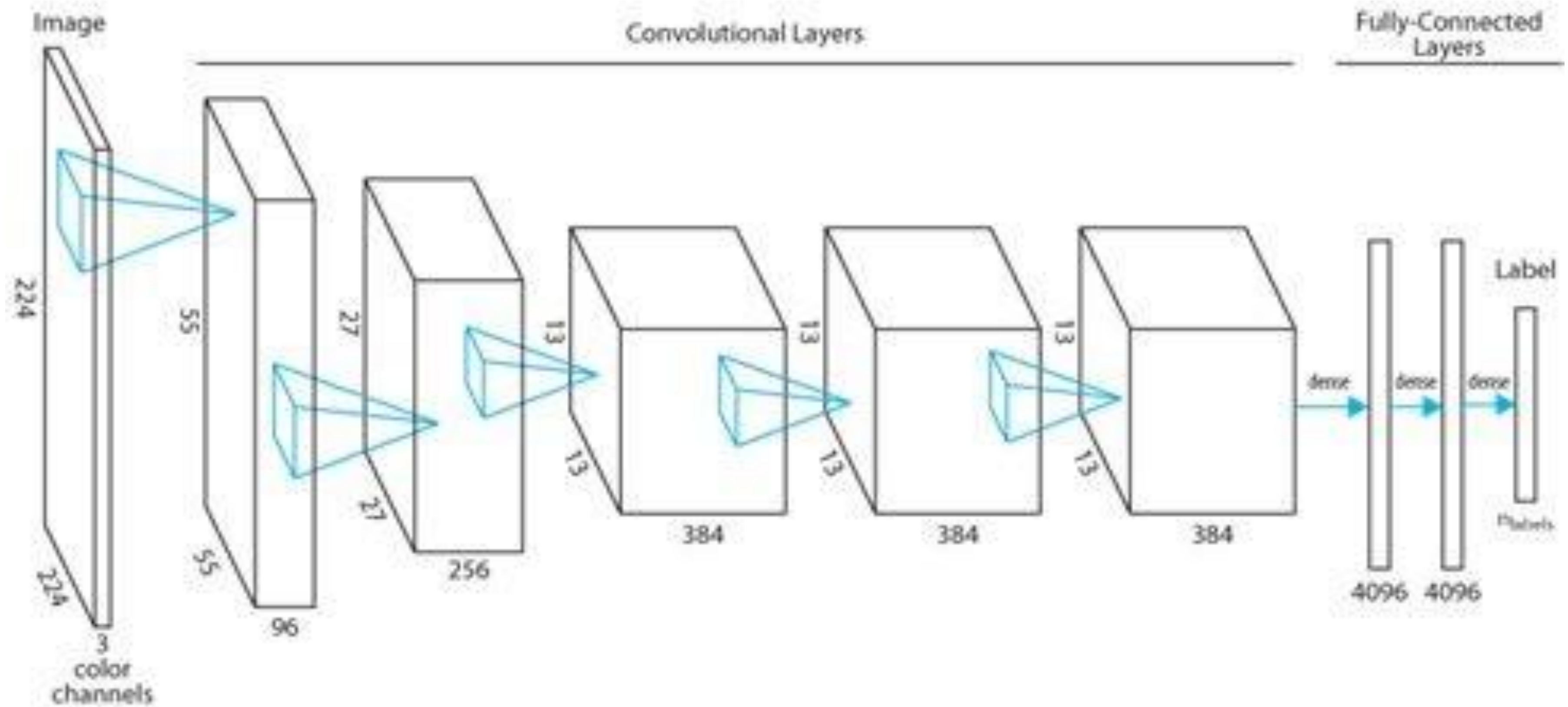
Remember Fully Connected Layers?





AlexNet - First Strong Result with CNNs

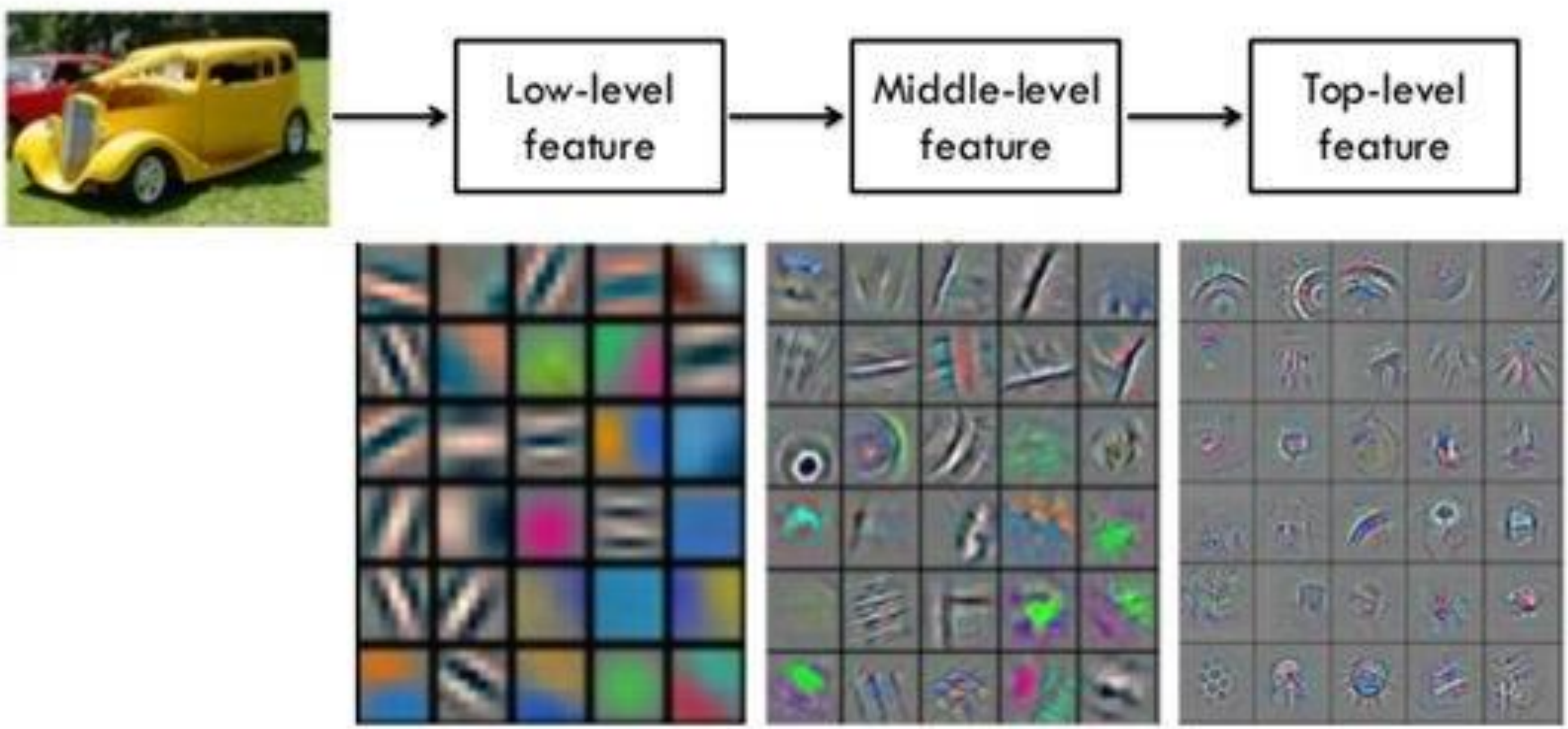
Alex Krizhevsky, Ilya Sutskever, Jeoffrey Hinton (2011)



- Same as LeNet, but more Convolutional layers.
- Dataset with bigger and more images (IMAGENET).
- Classify 1M images to 1000 categories.
- Implemented with modern GPUs leveraging high parallel processing capabilities.

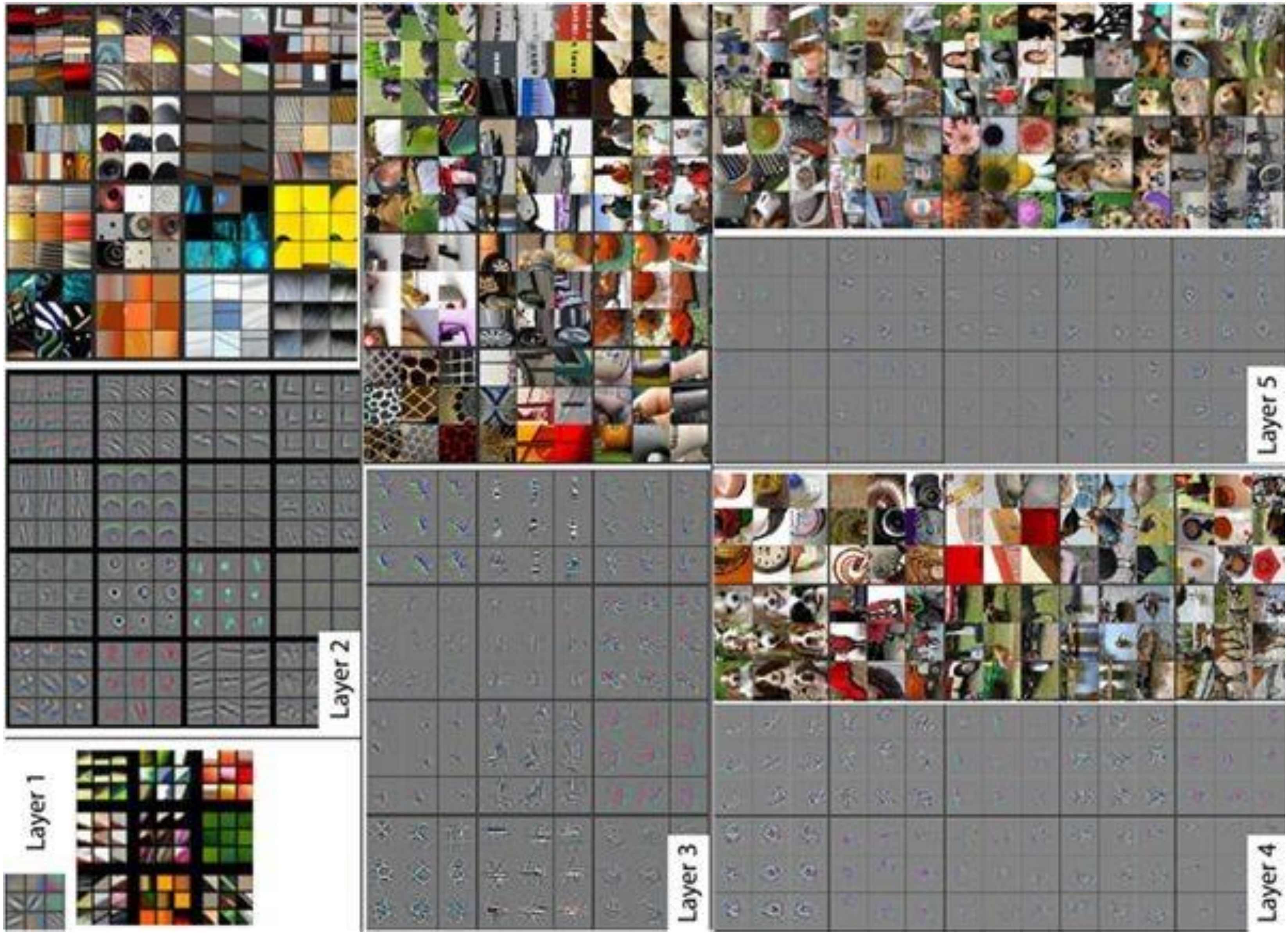
What CNNs Learns ?

Hierarchy of trained representations

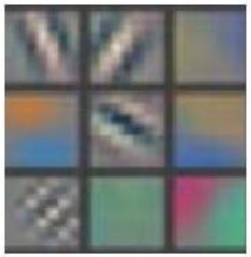


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

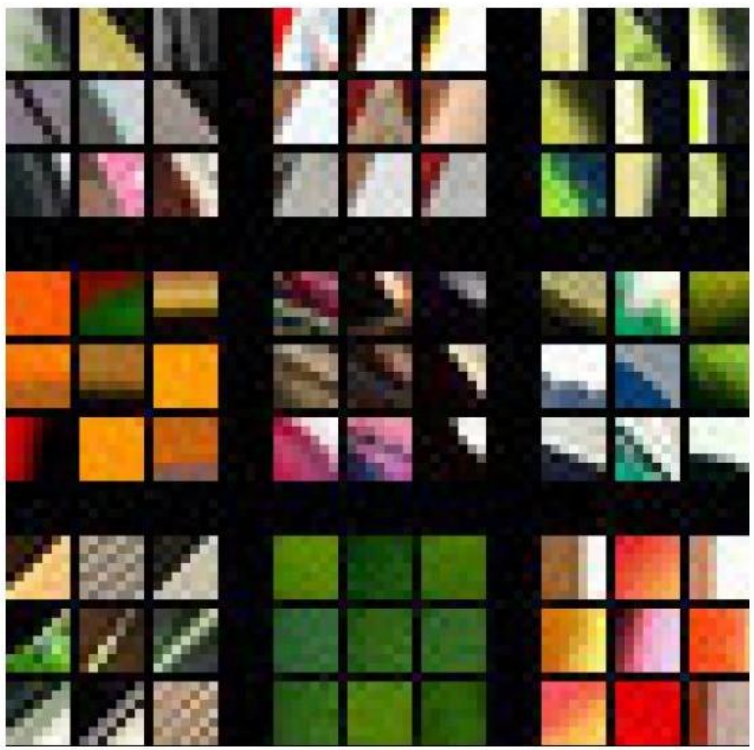
What AlexNet Learns?



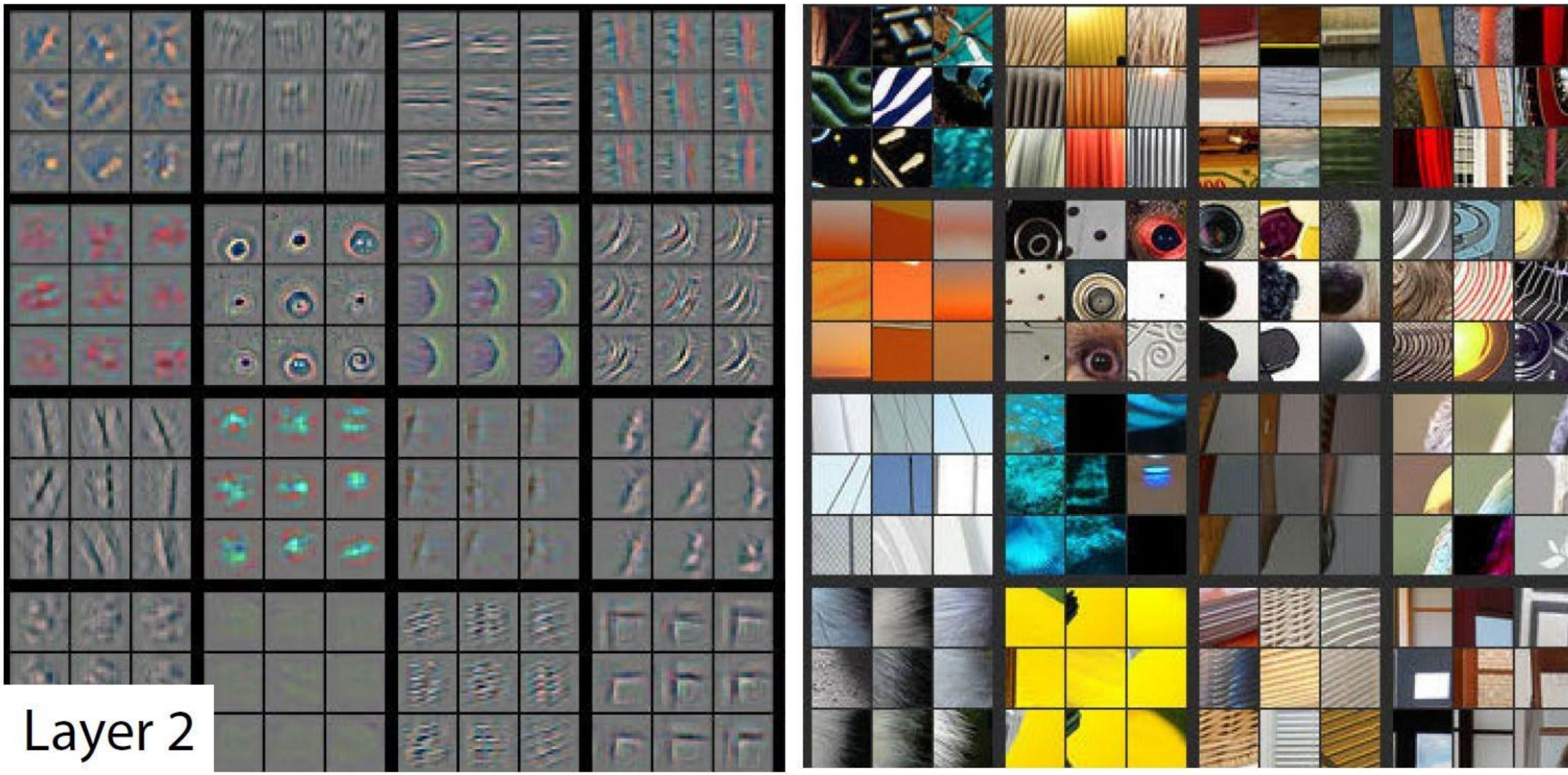
What AlexNet Learns?



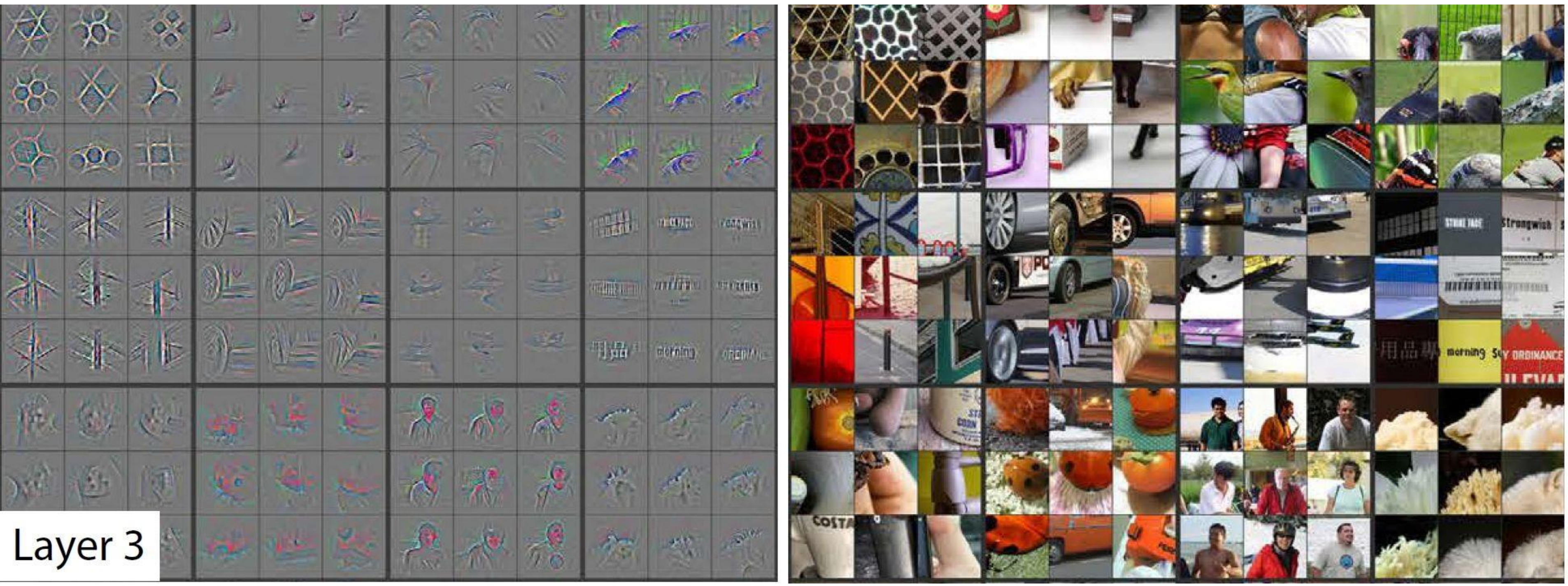
Layer 1



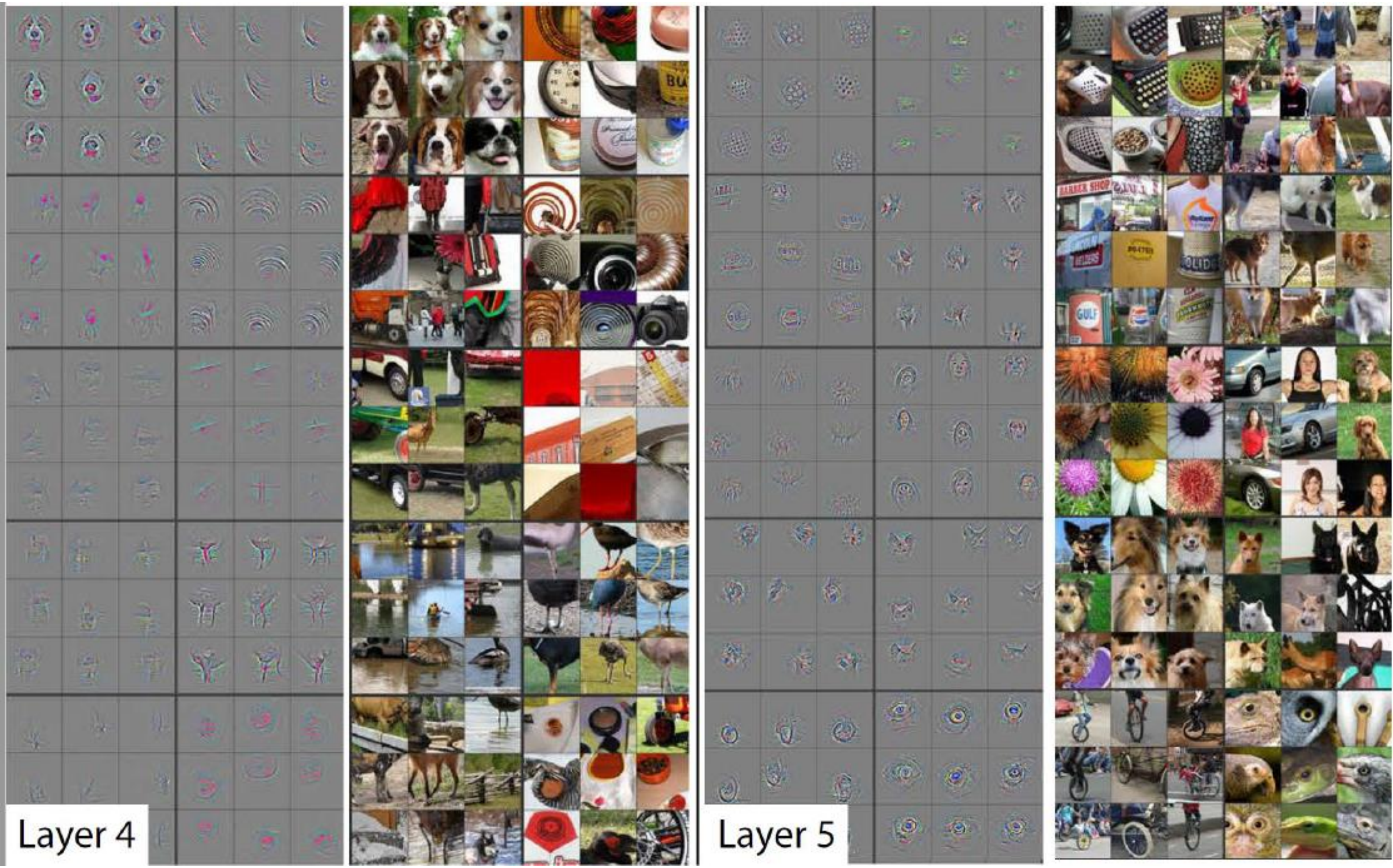
What AlexNet Learns?



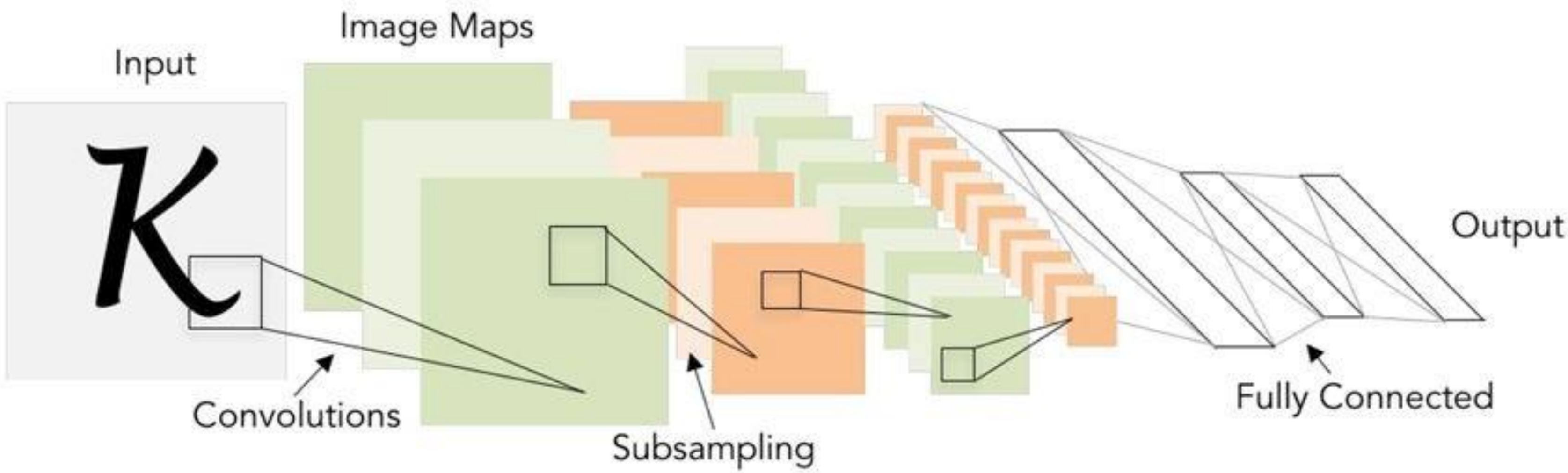
What AlexNet Learns?



What AlexNet Learns?



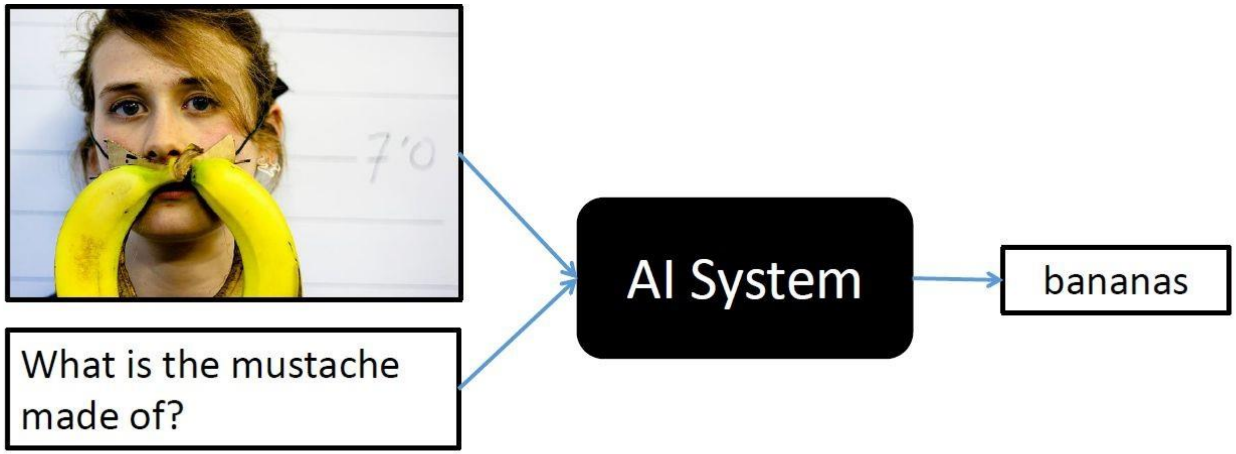
Deep Learning (CNNs) Recap



LeNet-5

- Think about input and output structure.
- Design a suitable network architecture - NN model.
  - Conv, Relu, Pooling, Fully connected Layers.
  - Deeper with few parameters.
- Define appropriate error (loss) function for learning.
- Minimize loss function to learn weights - backpropagation.

CNNs are Everywhere

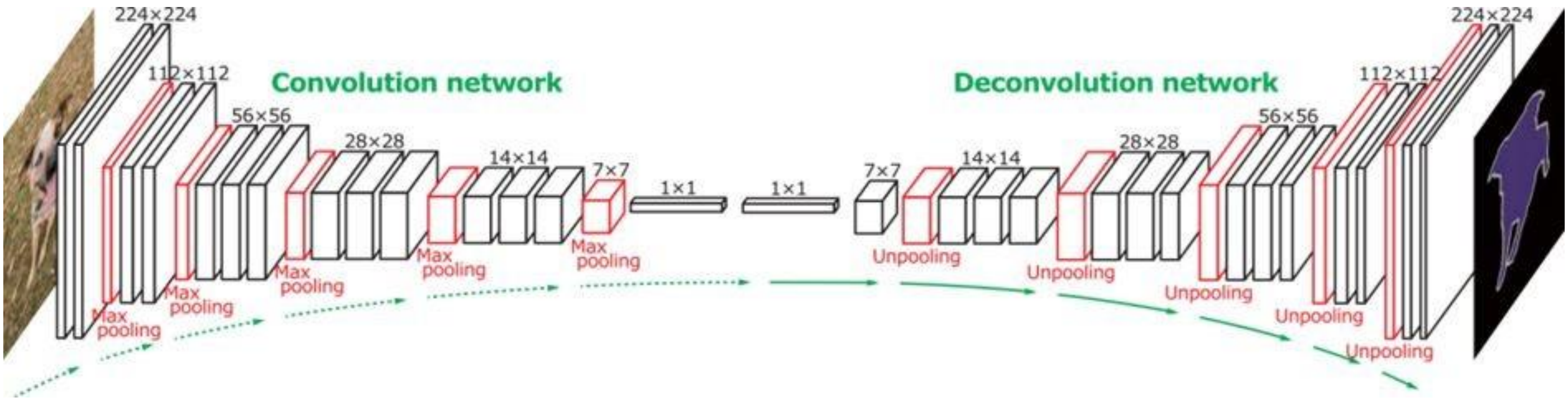


"man in black shirt is playing guitar."

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

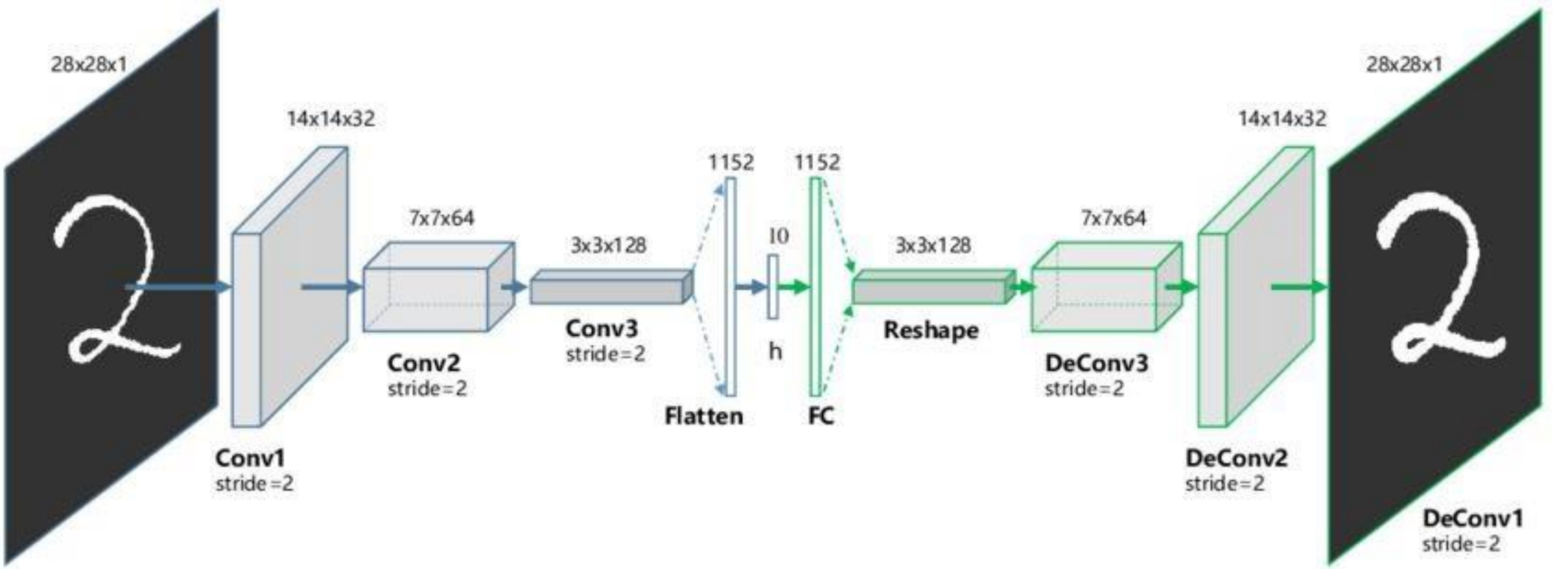
"two young girls are playing with lego toy."

CNNs with Structured Outputs (Image)



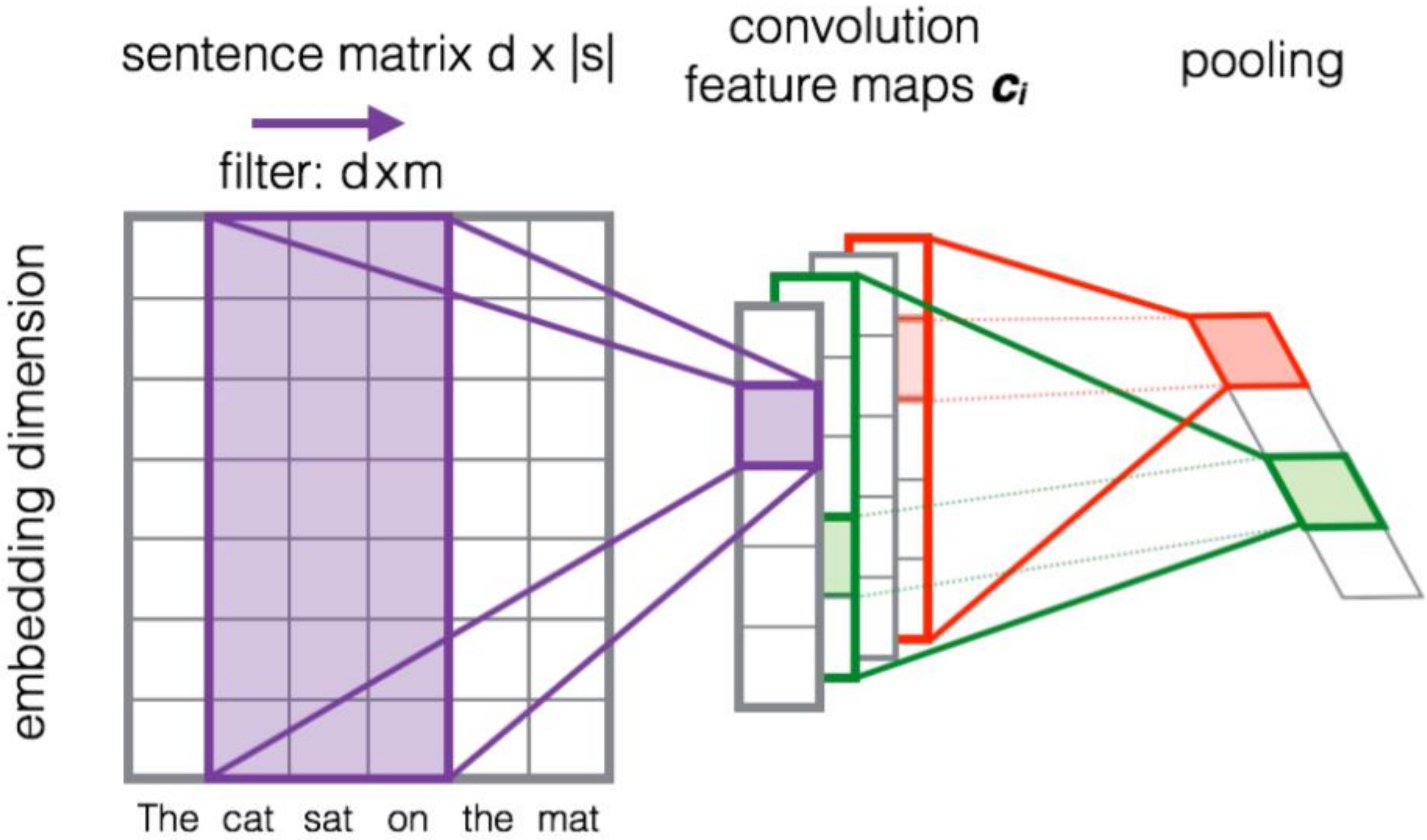
- Use successive Convolution - Downsampling to encode image.
- Deconvolution - Upsampling for decoding.
- Intuitively, invert the Convolutions and Subsampling operations.
    - Convolution to Transpose Convolution
    - Pooling to Unpooling

Unsupervised Deep Learning
Convolutional Autoencoder



- Can be used to learn abstract image representations as seen before **without classification labels**!!
- Can use reconstruction loss L1 or L2 difference in pixels.
- Image can be encoded to small vector which can be reused in intelligent decision making.

CNNs for Text Representation



[Severyn et al SIGIR'15]

Next UP

Implementing and Training CNNs