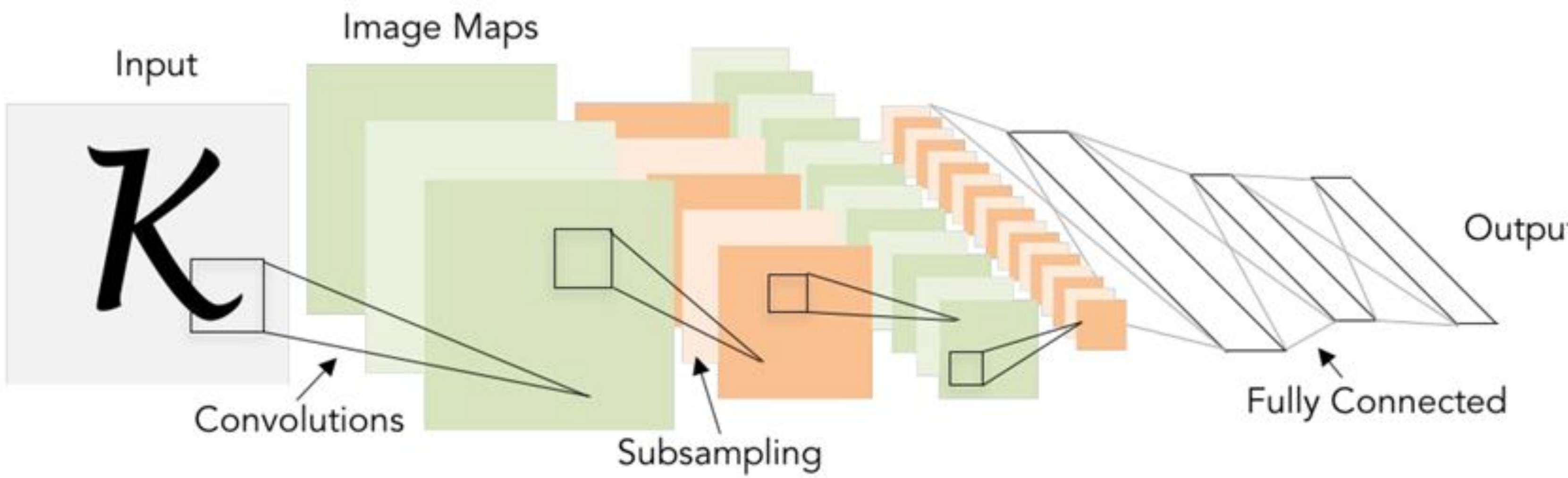
Convolutional Neural Networks

Artificial Intelligence

School of Computer Science
The University of Adelaide

Introduction

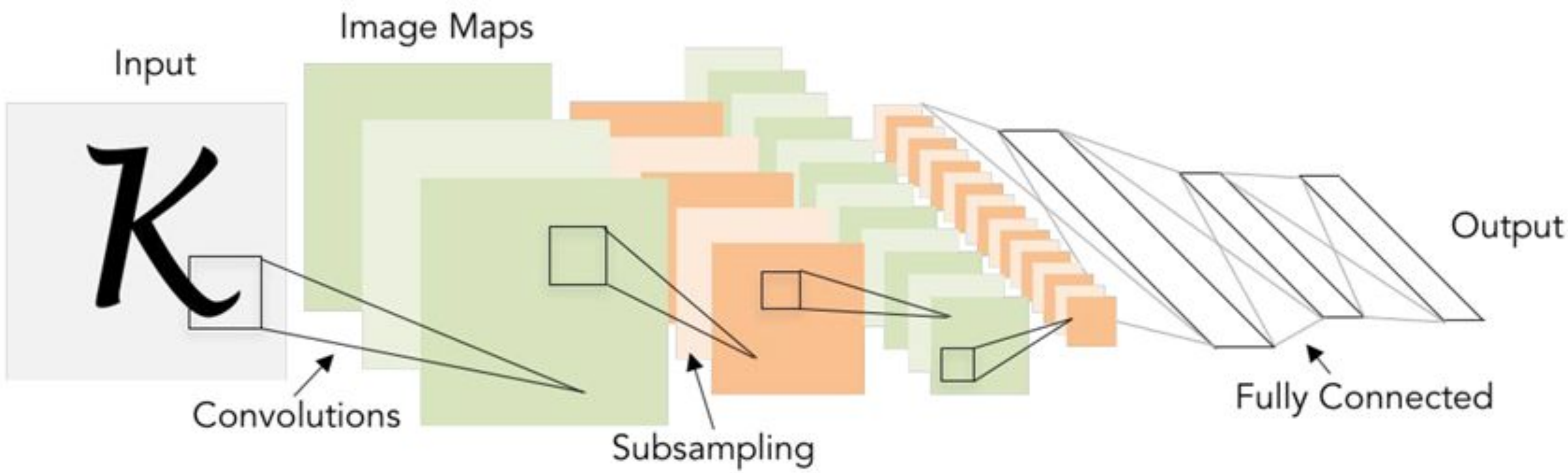
Introduced by Lecun etal. 1989.



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 with structured inputs.
  - e.g.images, videos, audio, graphical data ….

Building Blocks of Deep CNNs



LeNet-5

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

Convolution Operator (2D)

$$f * g[x,y] = \sum_{m=-M}^{M} \sum_{n=-N}^{N} f[x-m, y-n]\, g[m,n]$$

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\text{Image}} * \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}}_{\text{Filter}} = \underbrace{\begin{bmatrix} 2 & 4 & 2 \\ 3 & 6 & 3 \\ 2 & 4 & 2 \end{bmatrix}}_{\text{Filtered Image}}$$

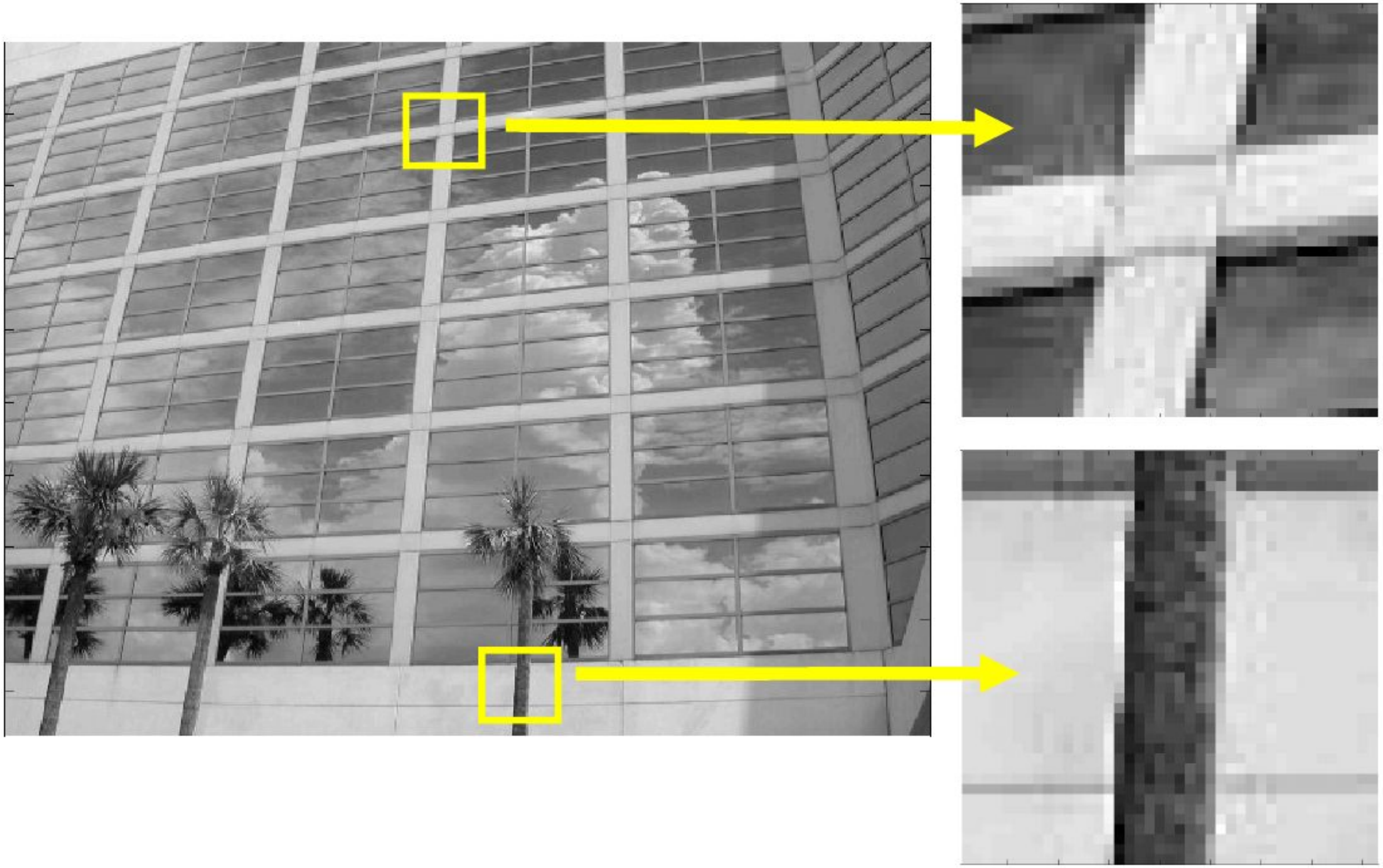
How is convolution done in practice?

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

$$
\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \cdots\cdots \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 \end{bmatrix} \cdots\cdots
$$

Convolution - Simple Pattern Detector

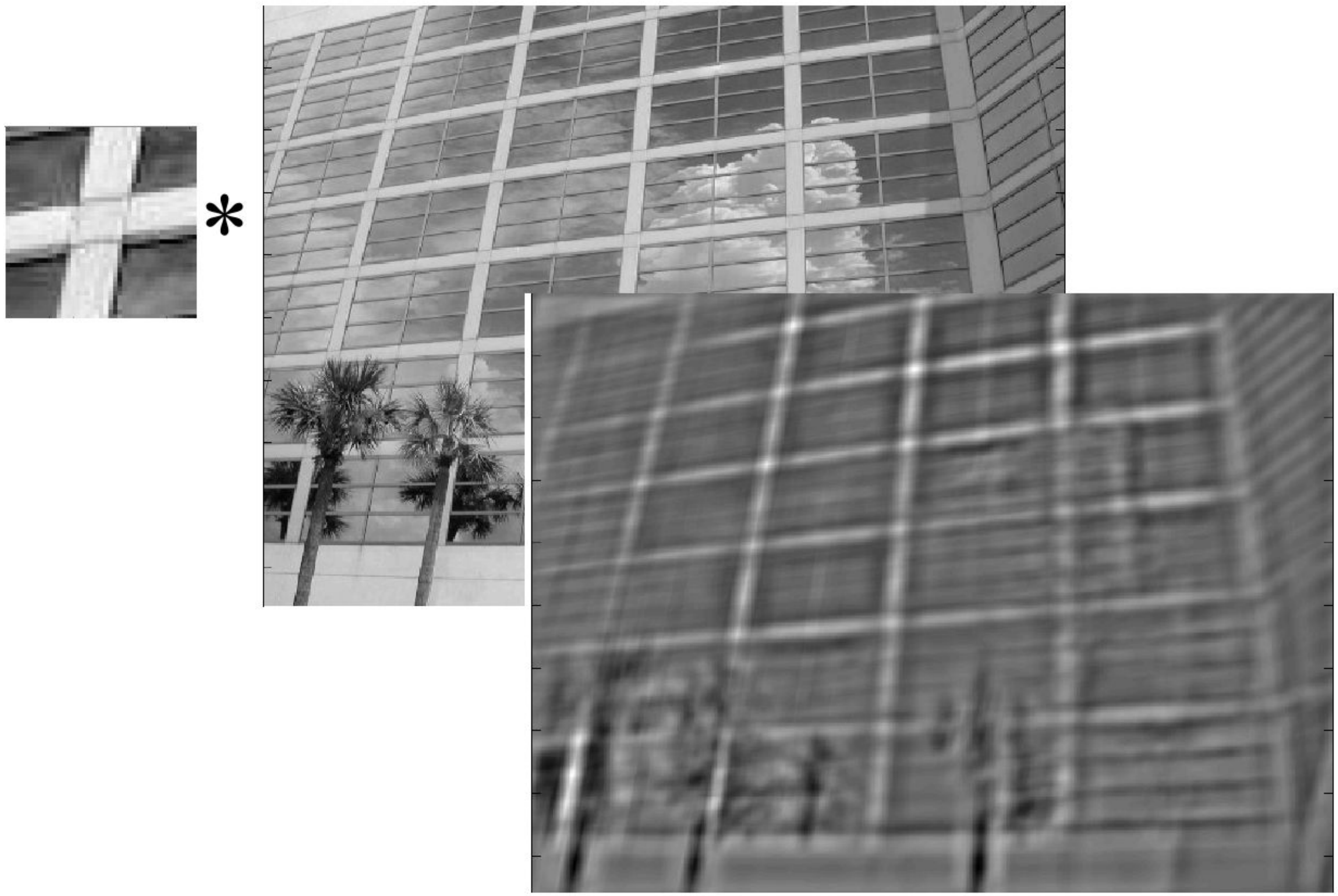
Convolving a filter with and image = detecting a template.



Robert Collins, CSE486, Penn State

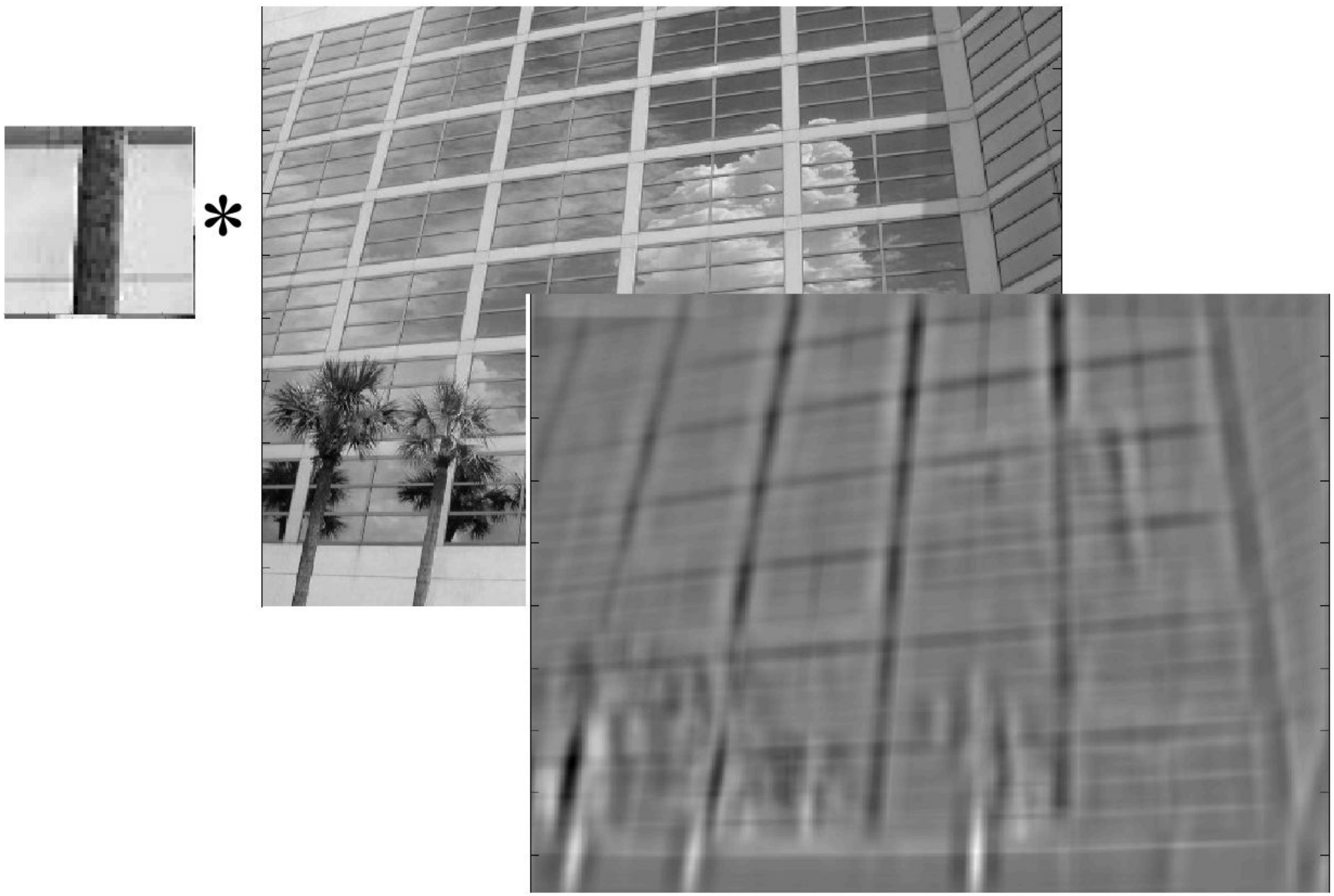
Convolution - Simple Pattern Detector

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



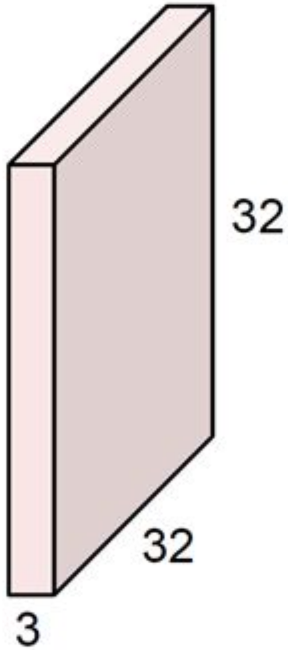
Convolution - Simple Pattern Detector

You can match multiple templates.

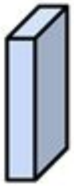


Convolutional Layer

32x32x3 image



5x5x3 filter

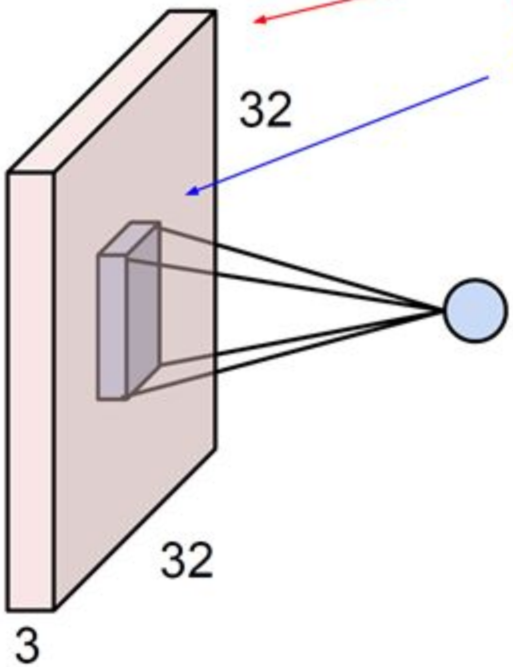


Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolutional Layer

32x32x3 image

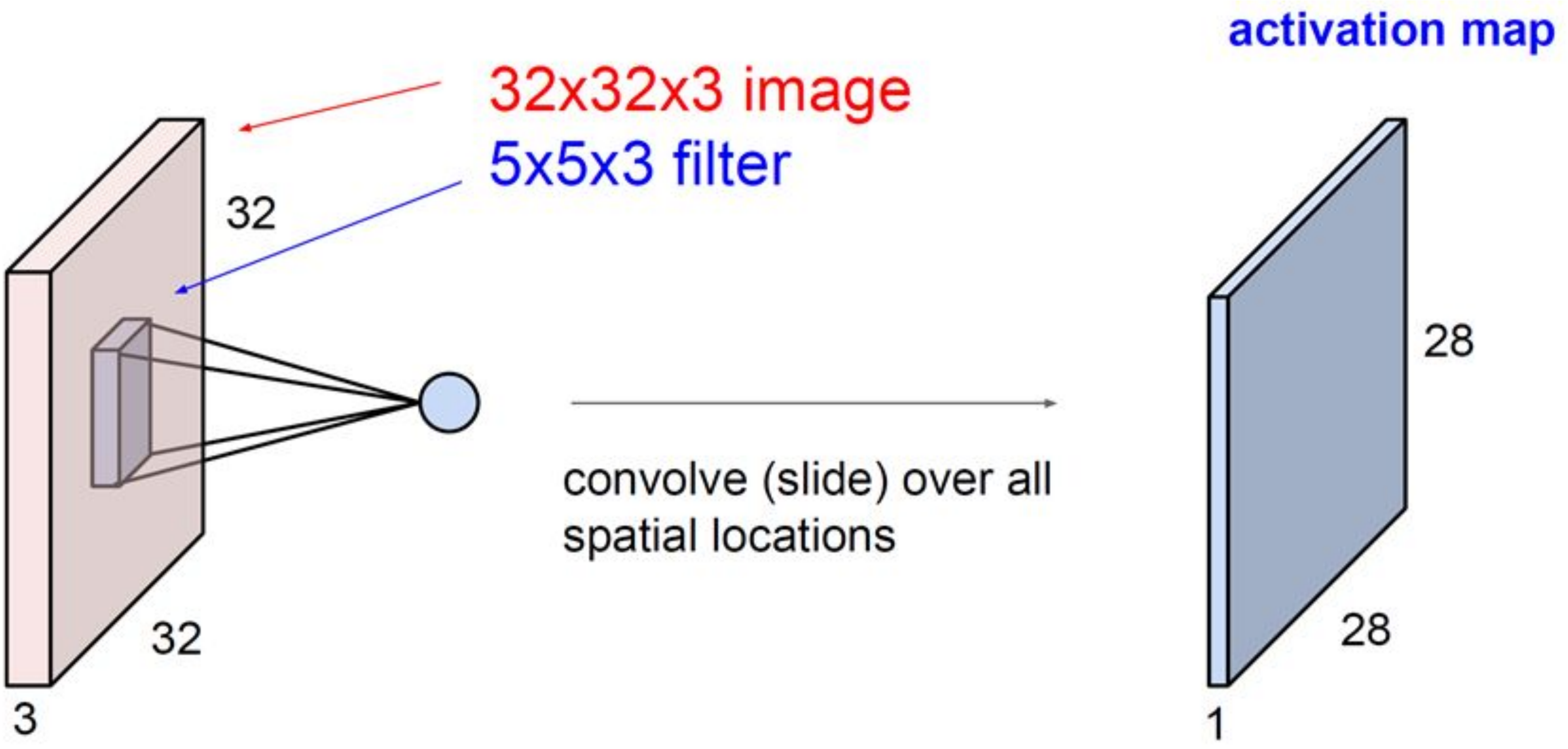
5x5x3 filter w



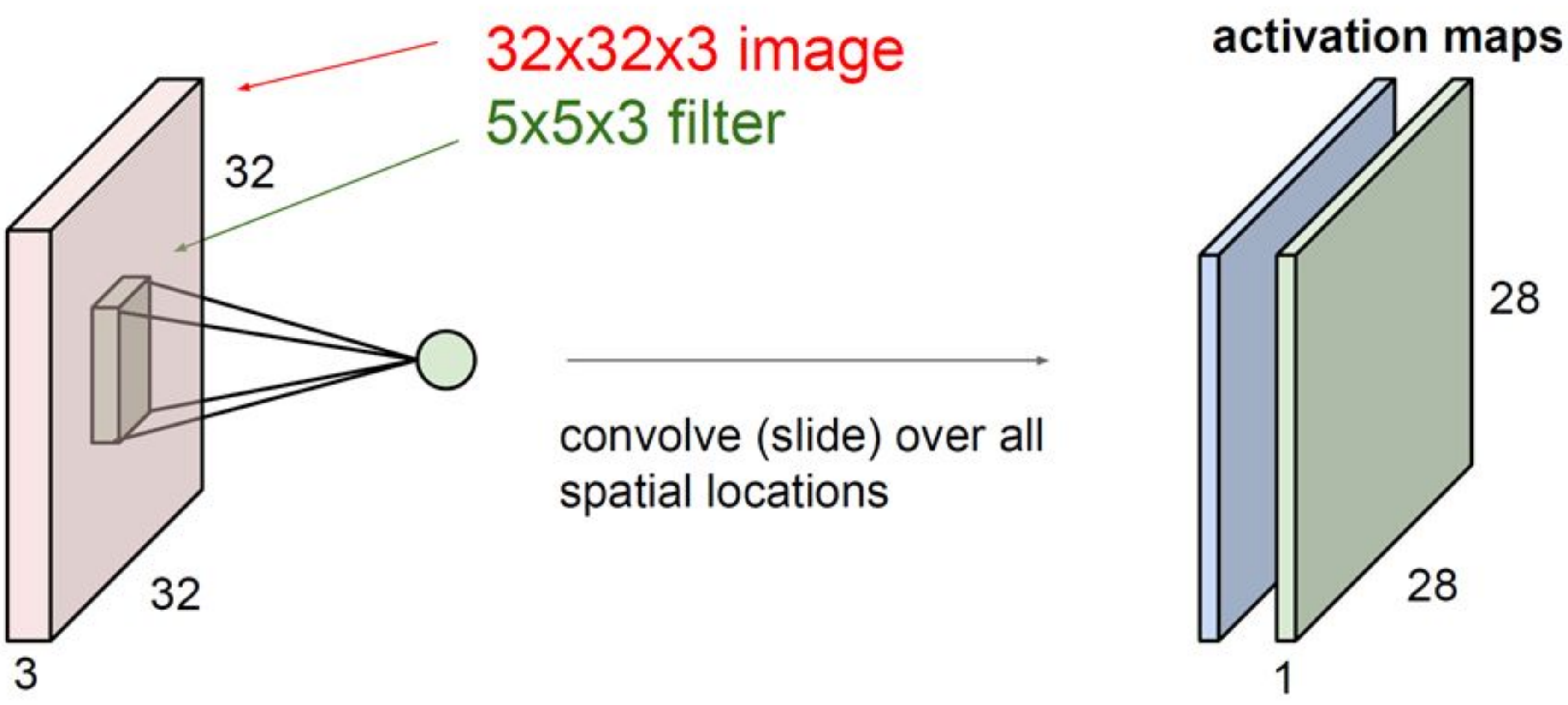
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

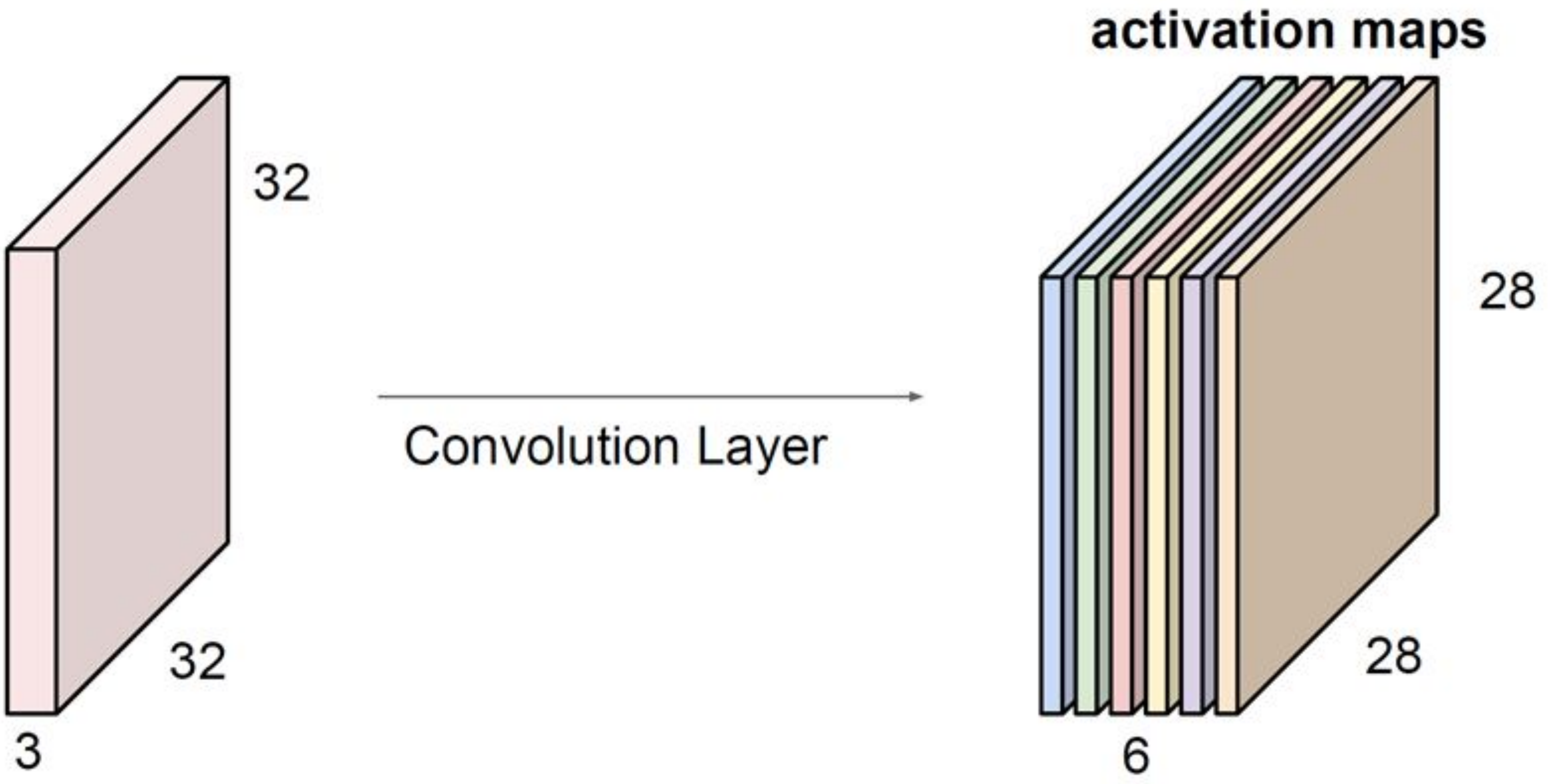
Convolutional Layer



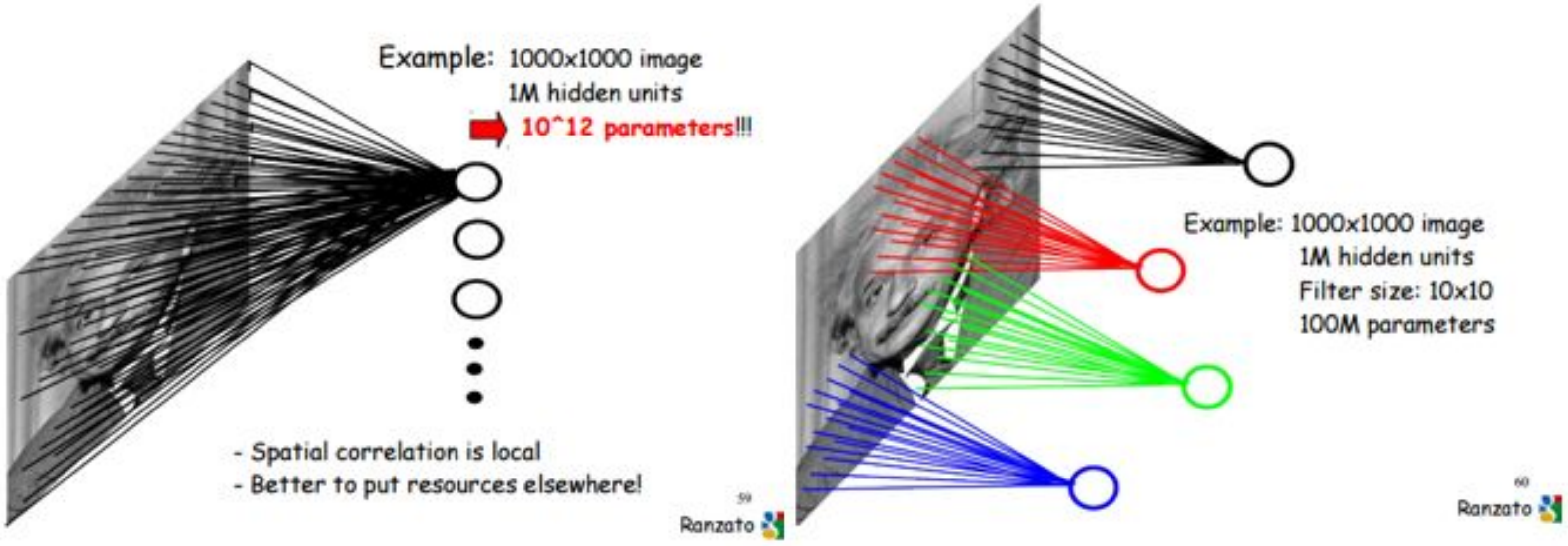
Convolutions More Filters



Convolution Layer

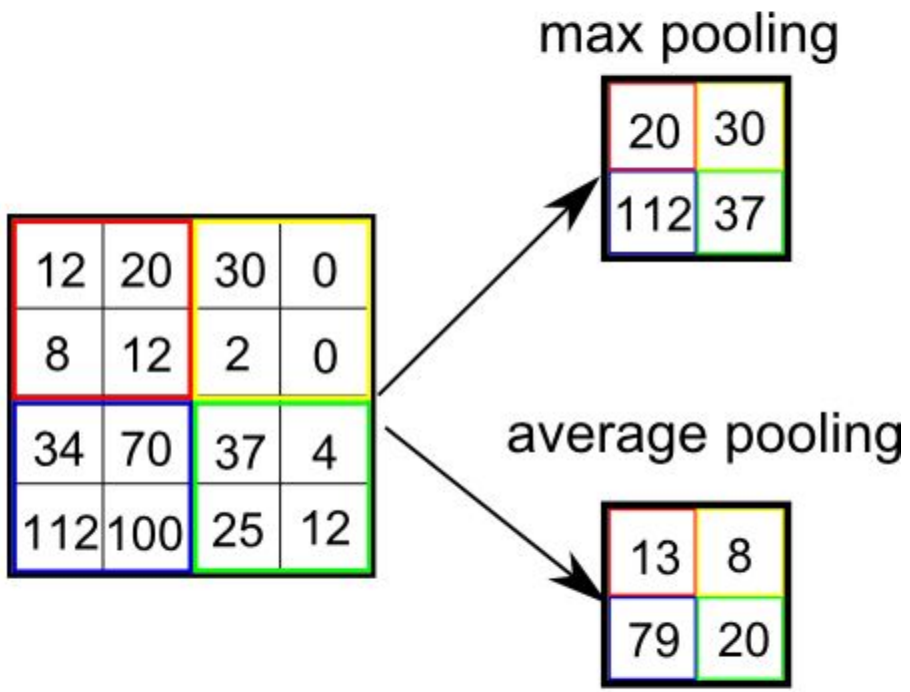


Why Convolutions?



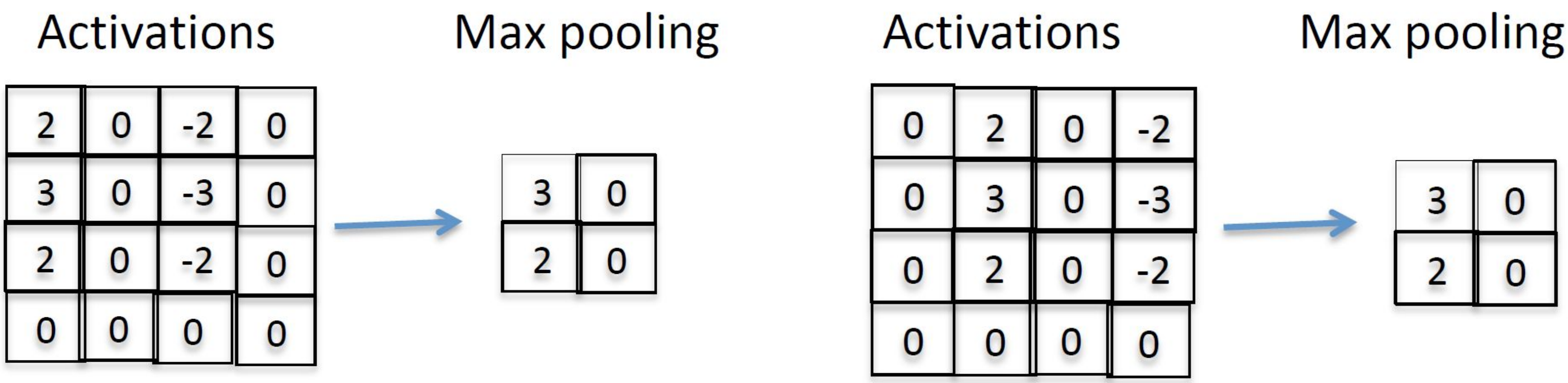
- Every output neuron has sparse connectivity - more tractable.
- **Weight Sharing** - detects repeated local structures in the data.
    - 1000 x 1000 image, 1M hidden units
    - 100 filters with size 10x10 (100 weights each)
    - 10k parameters

Subsampling - Pooling



- Max Pooling
- Average Pooling

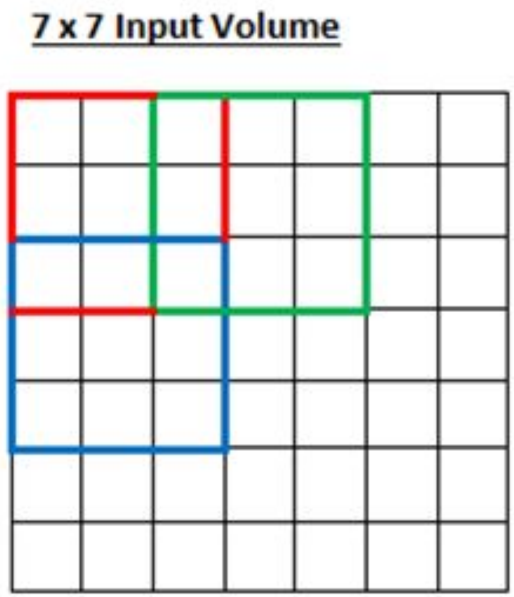
Subsampling - Pooling

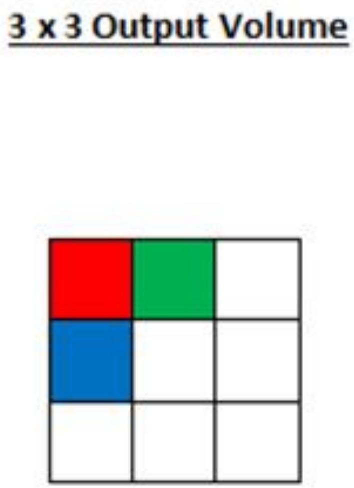


- Reduces size of the data and thus computation cost.
- Add translation invariance - Small horizontal or vertical translations does not affect the outputs.

Convolutions with Strides

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

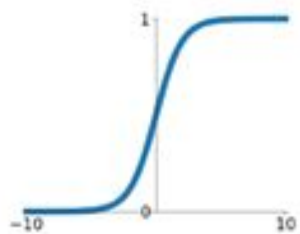




Remember Nonlinear Activations?

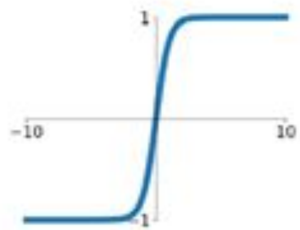
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



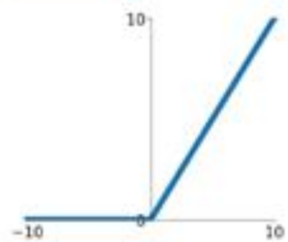
tanh

$\tanh(x)$



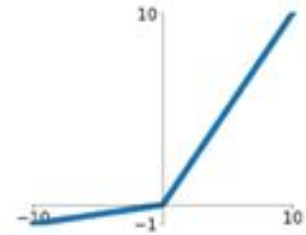
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

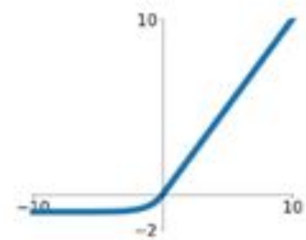


Maxout

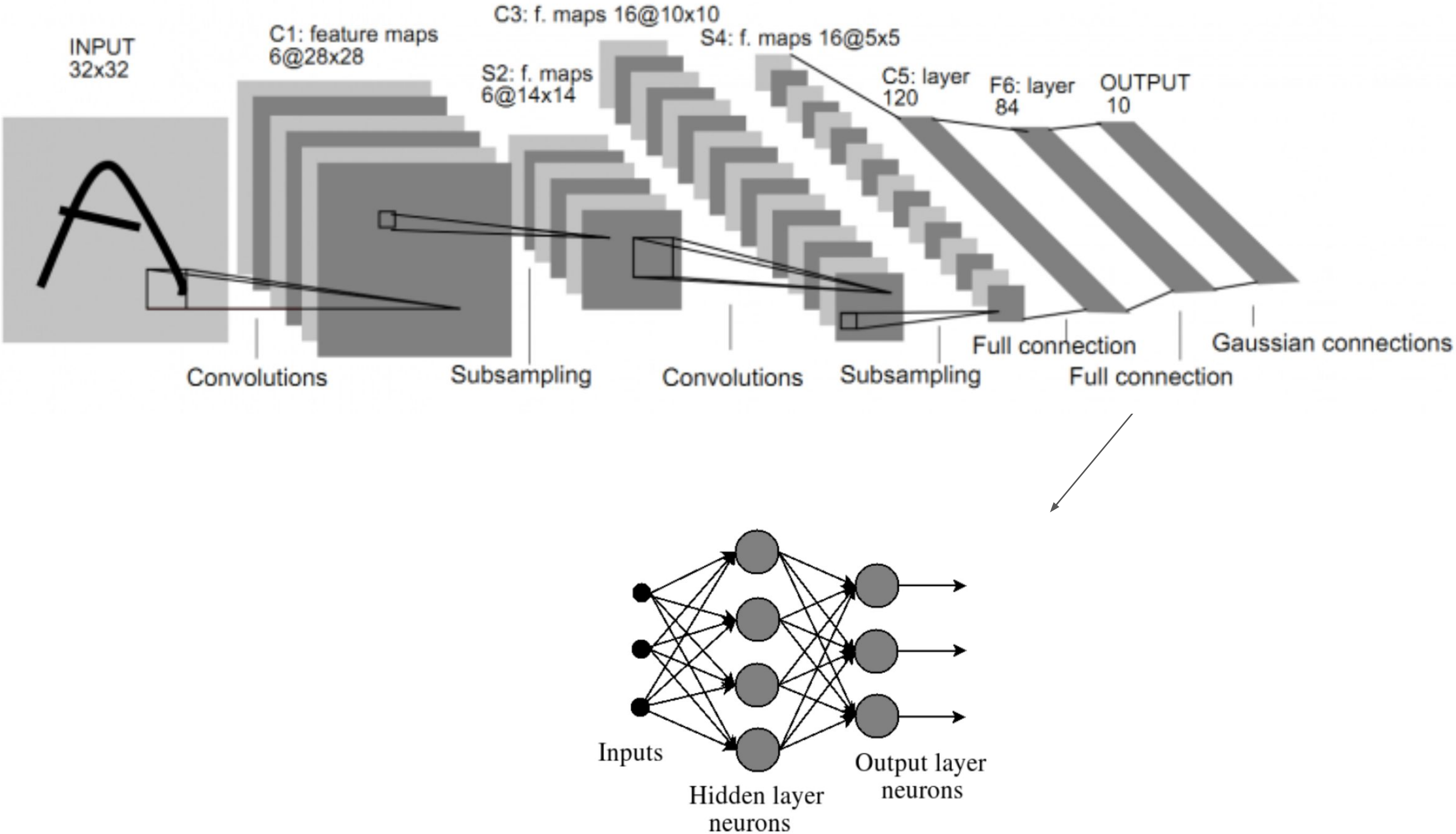
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

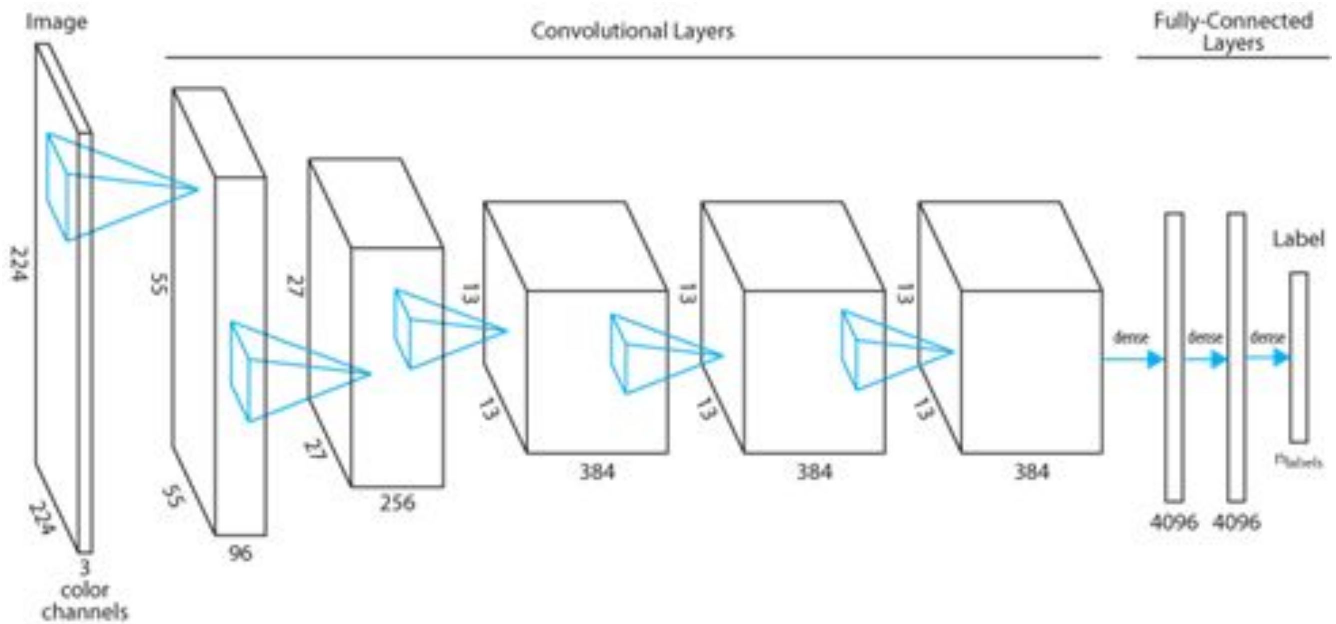
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



And Fully Connected Layers??

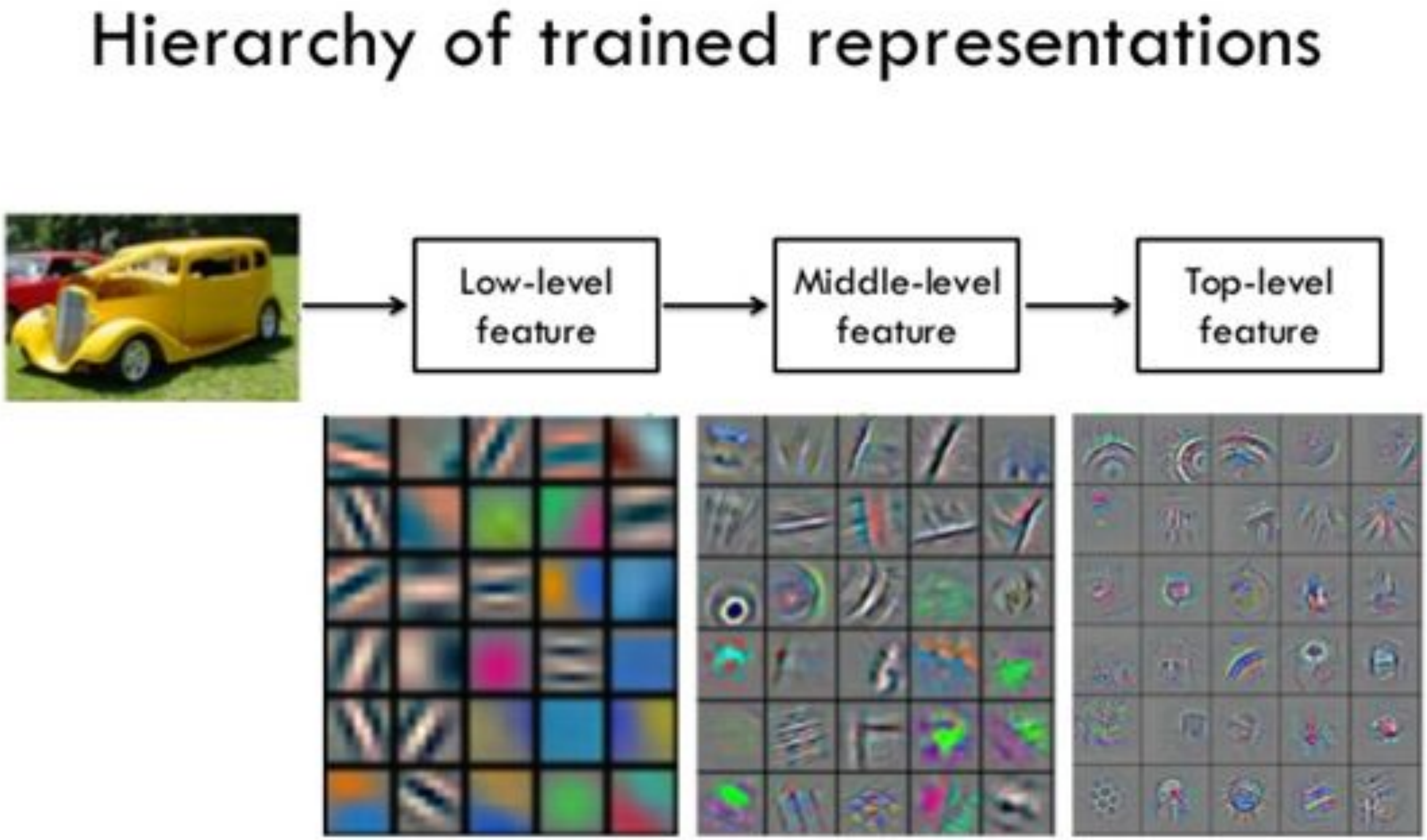


AlexNet - First Strong Result with CNNs
Alex Krizhevsky, Ilya Sutskever, Jeoffrey Hinton (2011)



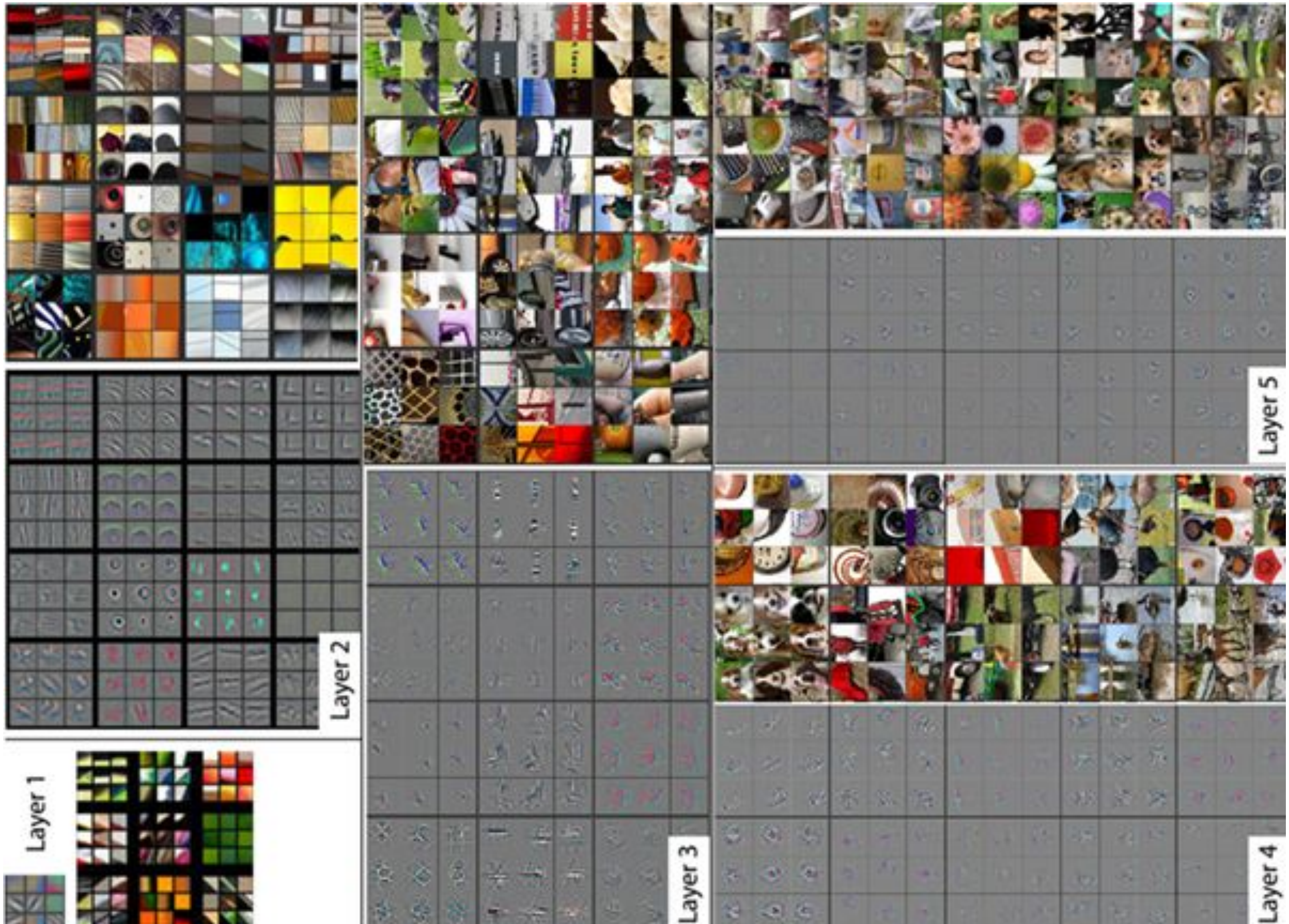
- Same as Lecun Net, but more Convolutional layers.
- Dataset with bigger and more images (IMAGENET).
- Classify 1M images to 1000 categories.
- Implemented with modern GPUs leveraging high parallel processing capabilities.

What CNNs Learns (Alexnet)?

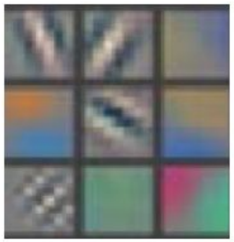


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

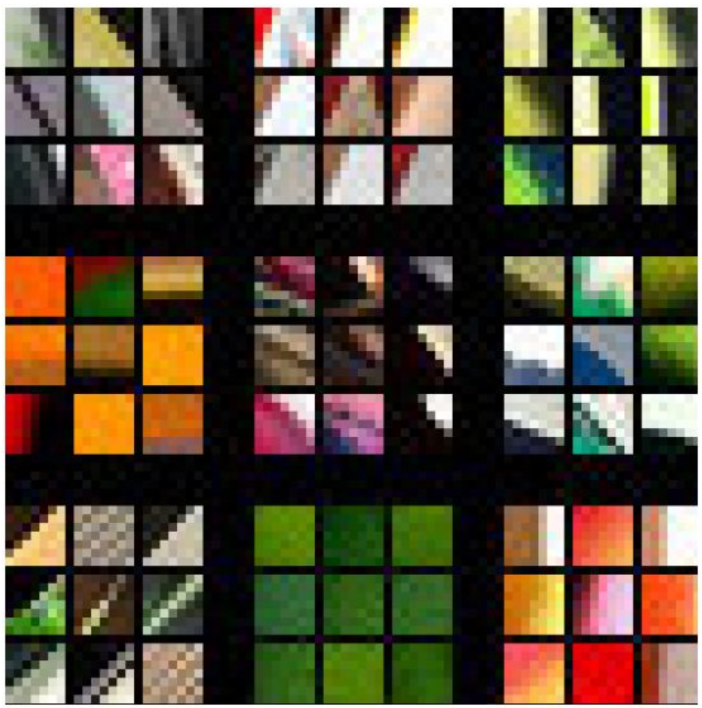
What AlexNet Learns?



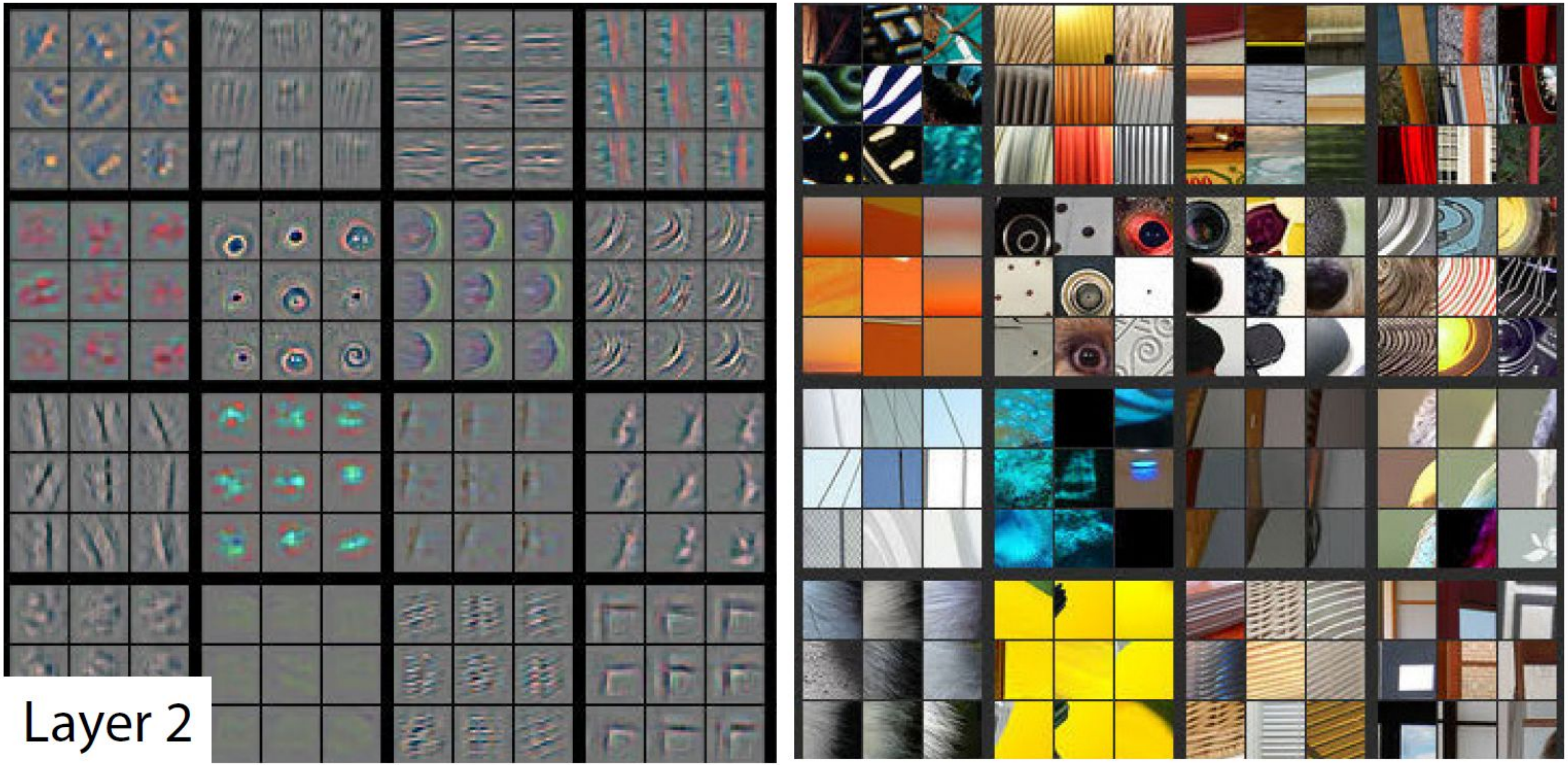
What AlexNet Learns?



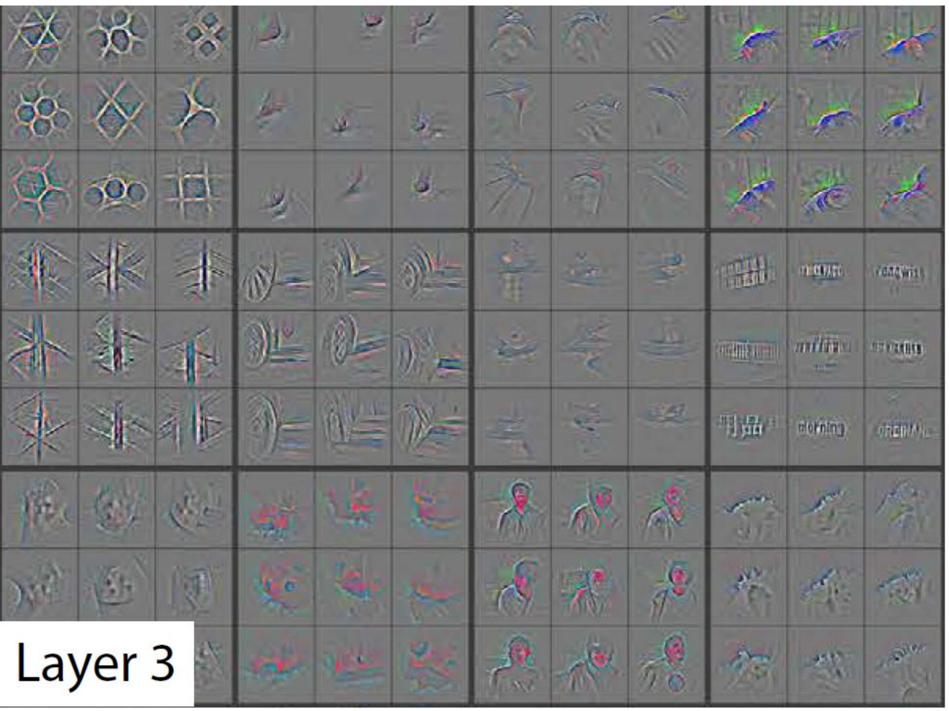
Layer 1



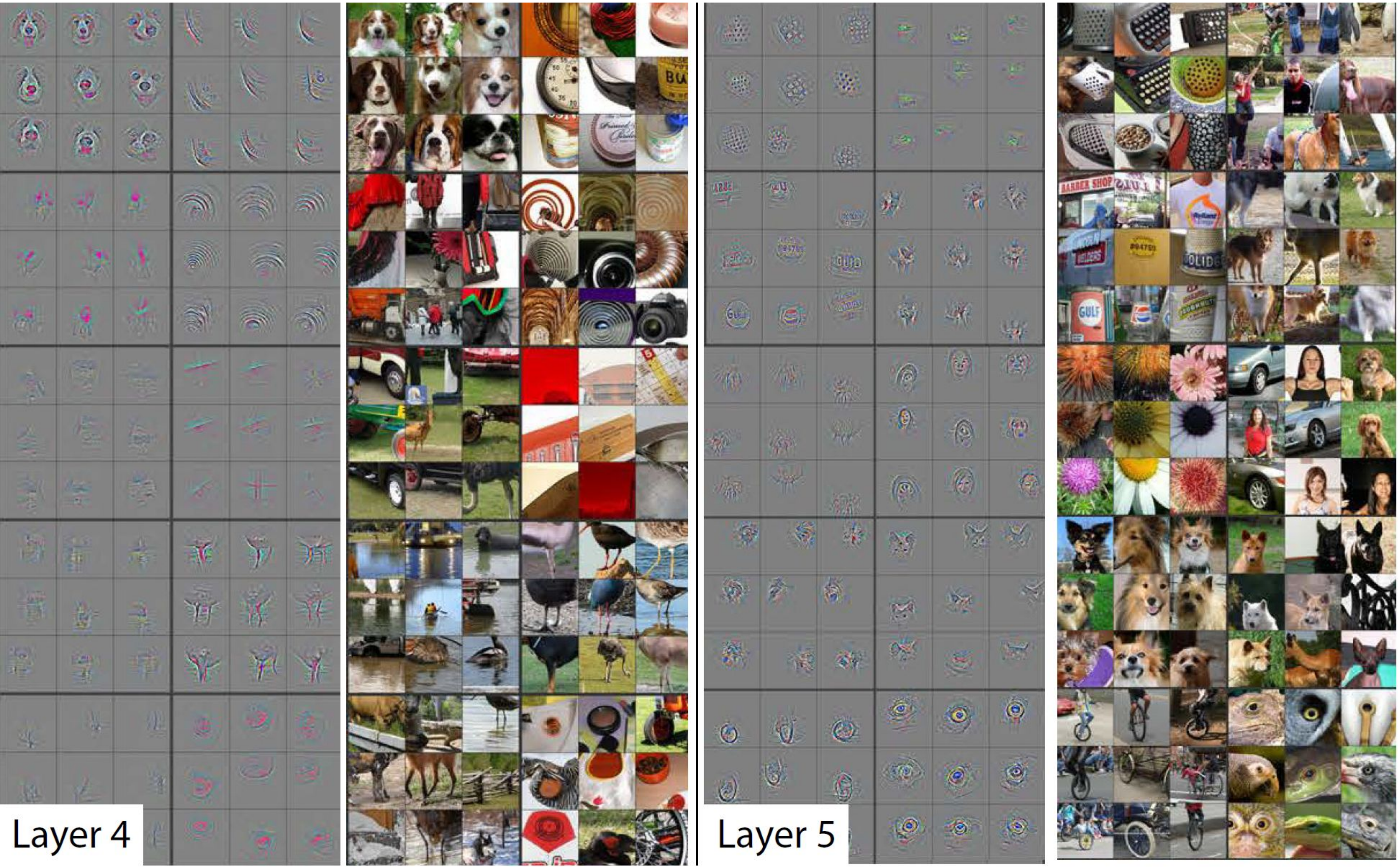
What AlexNet Learns?



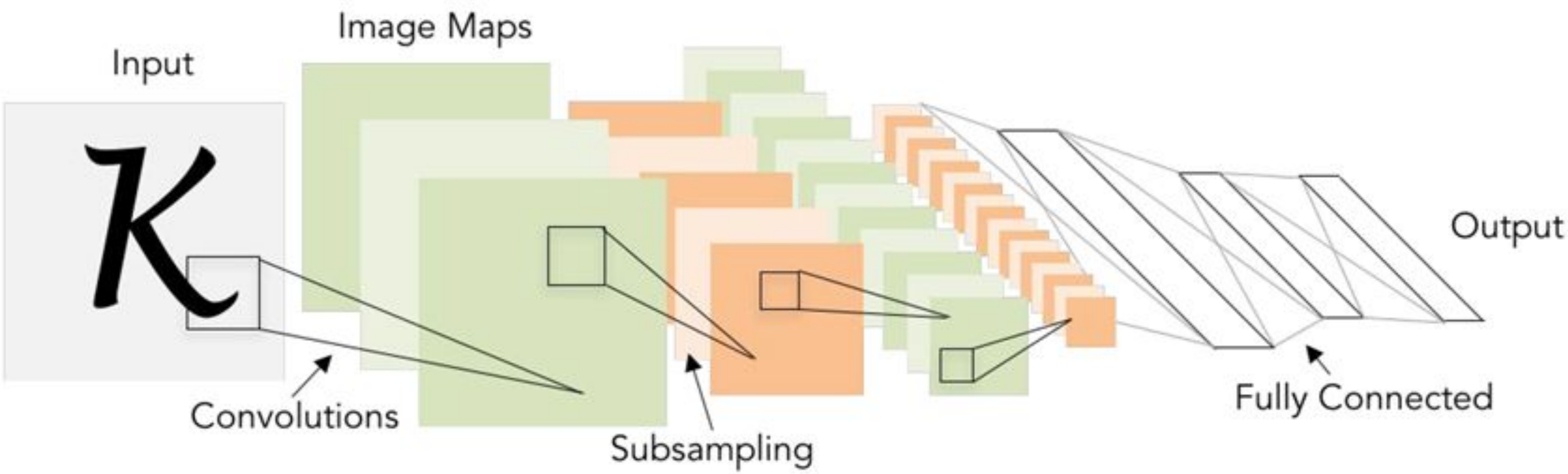
What AlexNet Learns?



What AlexNet Learns?



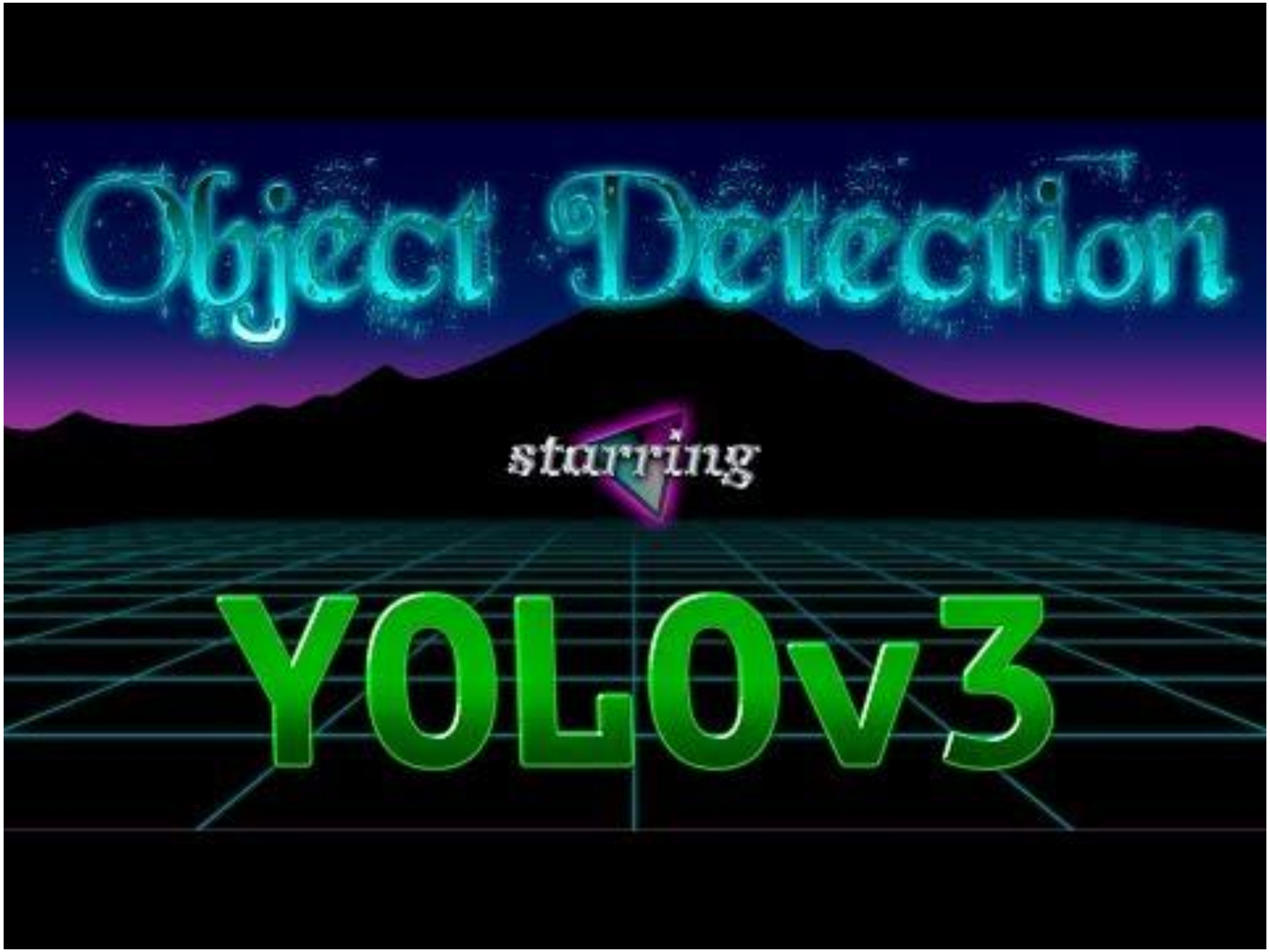
Deep Learning (CNNs) Recap



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.

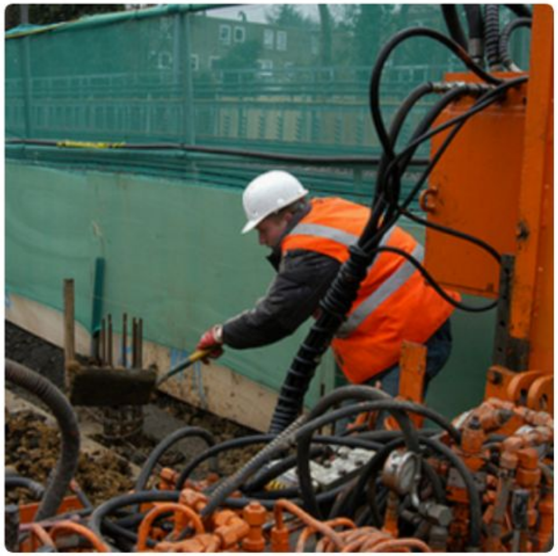
CNNs are Everywhere



CNNs are Everywhere

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."

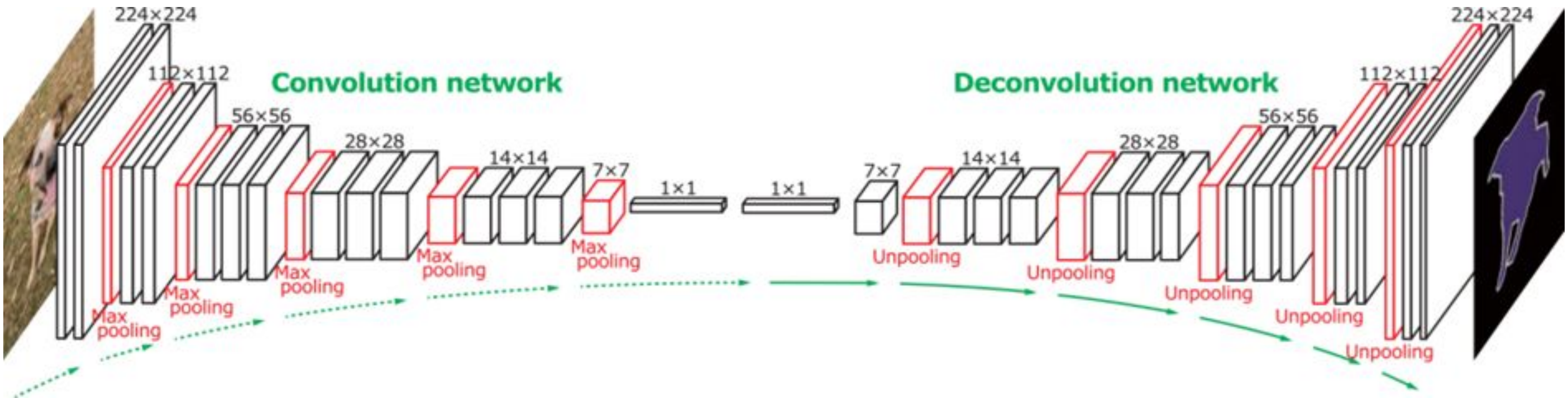
CNNs are Everywhere

What is the mustache made of?

AI System

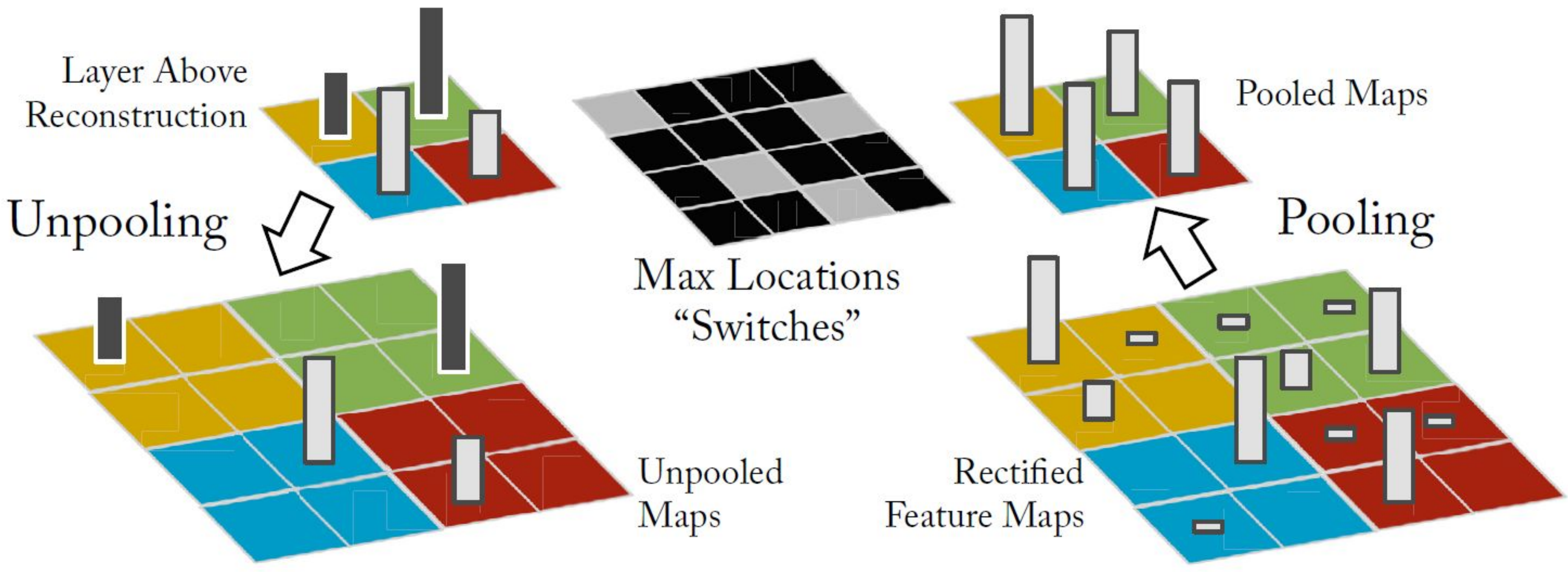
bananas

CNNs with Structured Outputs (Image)



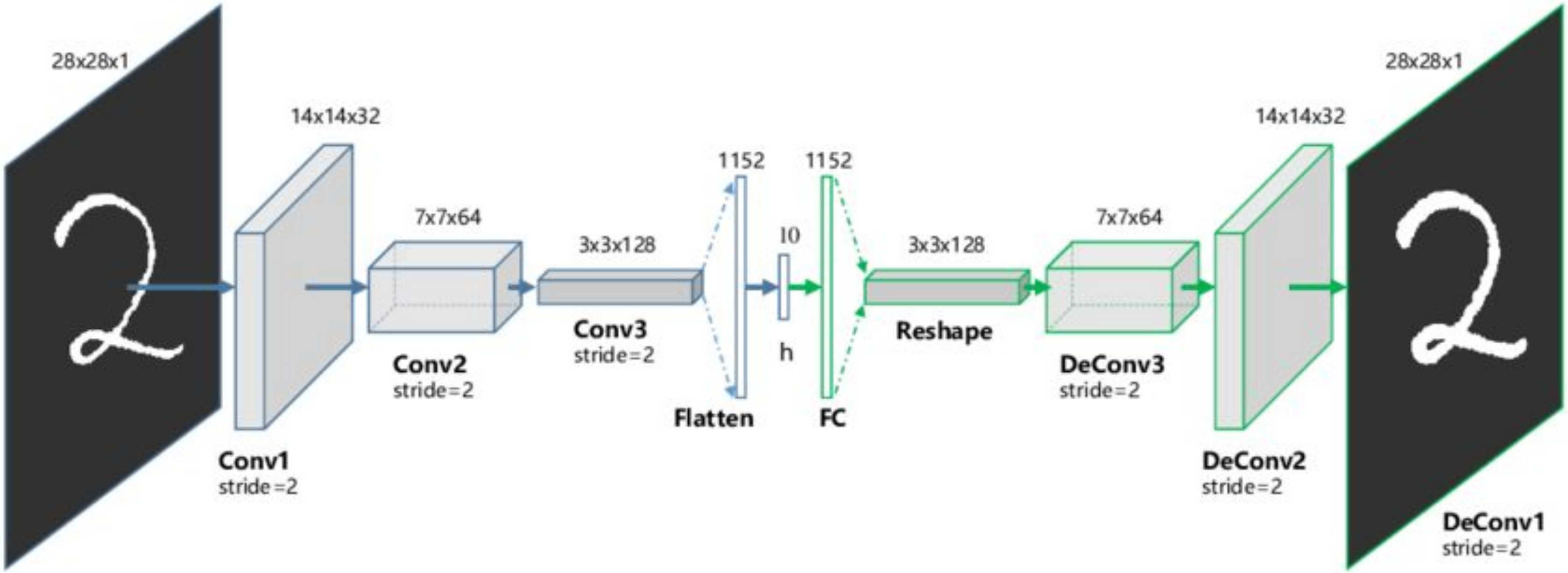
- Use successive Convolution - Downsampling to encode image.
- Deconvolution - Upsampling for decoding.
- Intuitively, invert the Convolutions and Subsampling operations.
  - Convolution to Transpose Convolution
  - Pooling to Unpooling

CNNs with Structured Outputs



- Intuitively, reverse 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.

Next UP

Implementing and Training CNNs