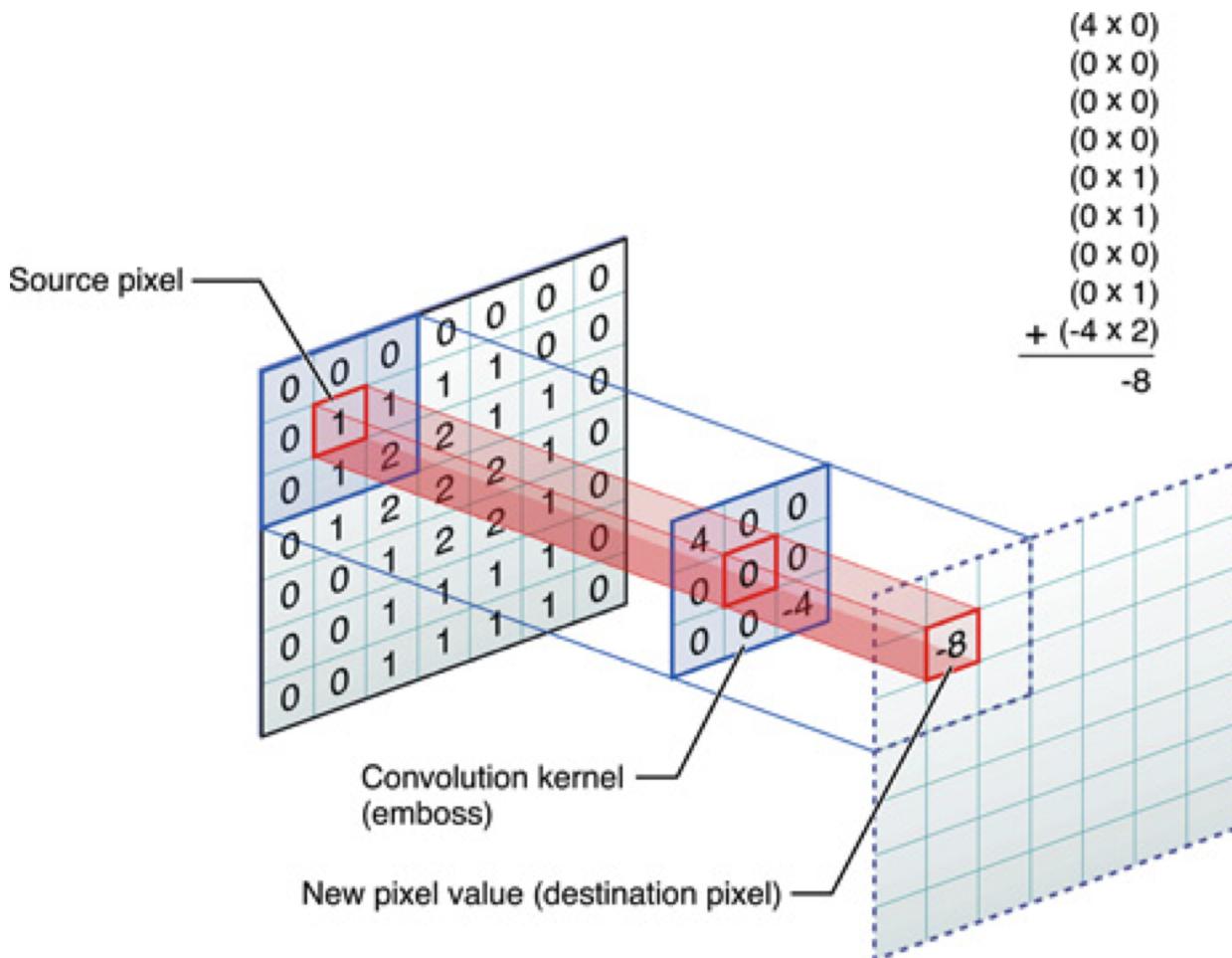


Big Data Technology

Paul Rad, Ph.D.

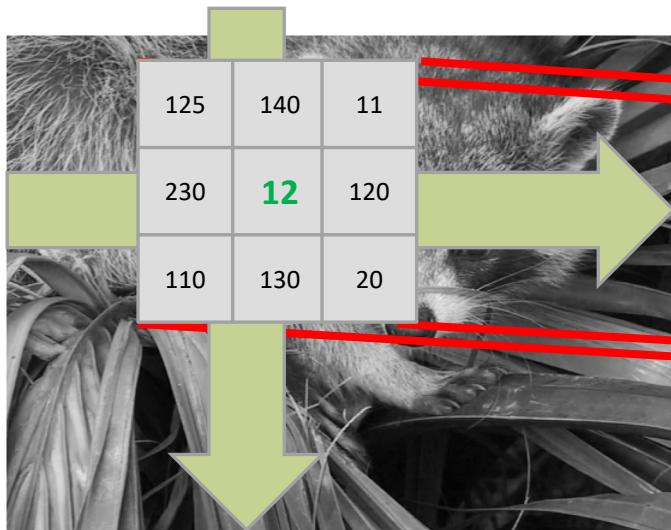
Associate Professor
Information Systems and Cyber Security, College of Business School
Electrical and Computer Engineering, College of Engineering

2-Dimensional Convolution



Example: 2-Dimensional Convolution

A convolution is an integral (**discrete signals :Matrix Dot Product**) that expresses the amount of overlap of one function as it is shifted over another function



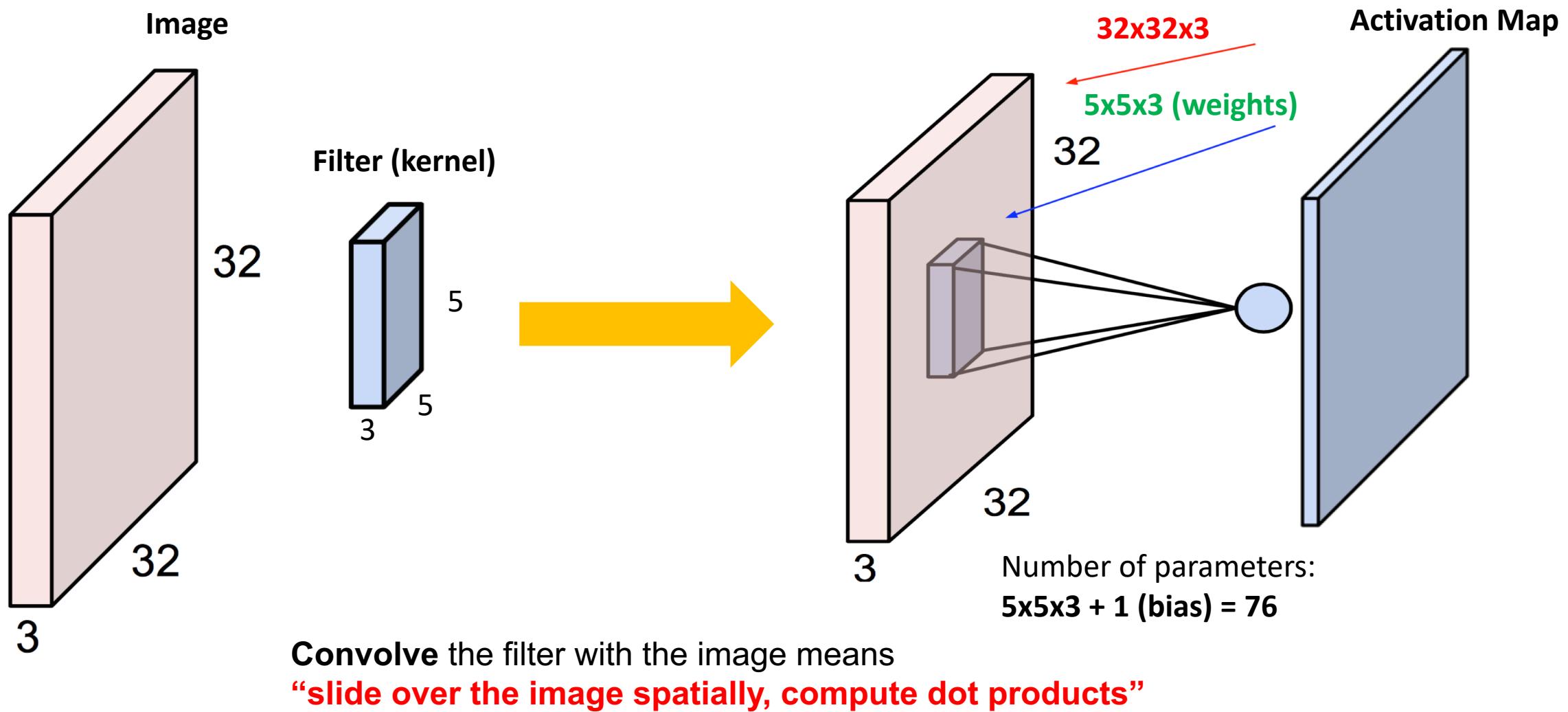
-1	-2	-1
0	0	0
1	2	1



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



Convolution Layer



Input Volume (+pad 1) (7x7x3)

 $x[:, :, 0]$

0	0	0	0	0	0	0	0
0	2	0	1	1	0	0	0
0	2	1	2	2	1	0	0
0	2	0	0	1	2	0	0
0	1	1	2	2	1	0	0
0	0	1	0	2	2	0	0
0	0	0	0	0	0	0	0
$x[:, :, 1]$	0	0	0	0	0	0	0
0	1	2	1	1	2	0	0
0	1	2	1	2	0	0	0
0	2	0	1	2	2	0	0
0	2	2	2	1	0	0	0
0	0	1	0	2	2	0	0
0	0	0	0	0	0	0	0
$x[:, :, 2]$	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0
0	1	1	1	0	2	0	0
0	2	1	1	2	1	0	0
0	0	2	1	1	0	0	0
0	0	0	2	1	2	0	0
0	0	0	0	0	0	0	0

Filter W0 (3x3x3)

 $w0[:, :, 0]$

1	1	1
1	1	1
0	-1	0

 $w0[:, :, 1]$

1	0	1
-1	1	0
-1	1	1

 $w0[:, :, 2]$

1	0	1
-1	1	0
-1	1	1

 $b0 (1x1x1)$ $b0[:, :, 0]$

1

Filter W1 (3x3x3)

 $w1[:, :, 0]$

1	1	0
0	0	-1
0	0	1

 $w1[:, :, 1]$

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

 $w1[:, :, 2]$

1	1	-1
-1	1	1
1	1	0

 $b1 (1x1x1)$ $b1[:, :, 0]$

0

Output Volume

 $o[:, :, 0]$

9
2

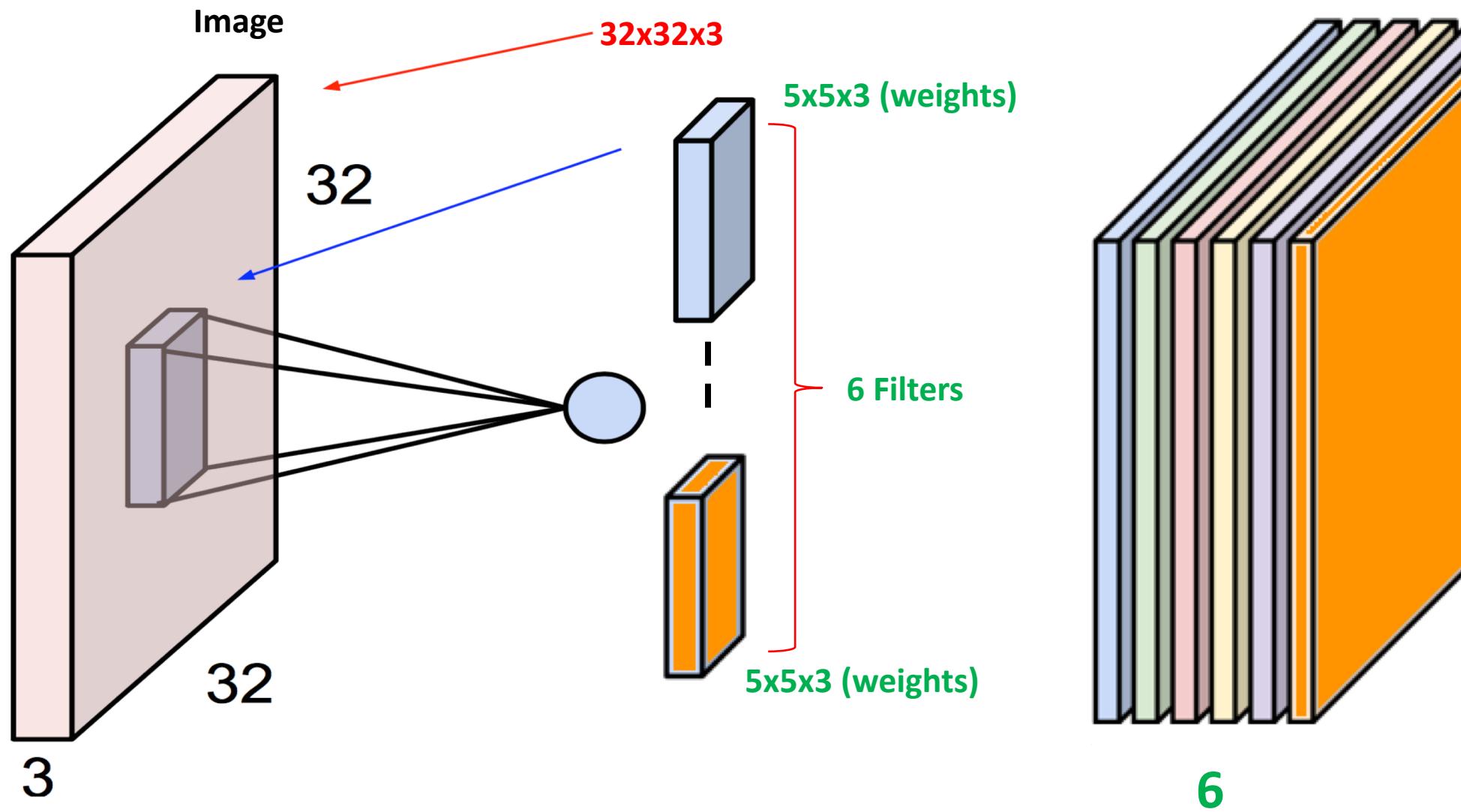
 $o[:, :, 1]$ 2

$$(2 \times 1) + (1 \times 1) + 0 + (1 \times 1) + (2 \times 1) + 0 + (2 \times 1) + (1 \times 1) + 0 = 9$$

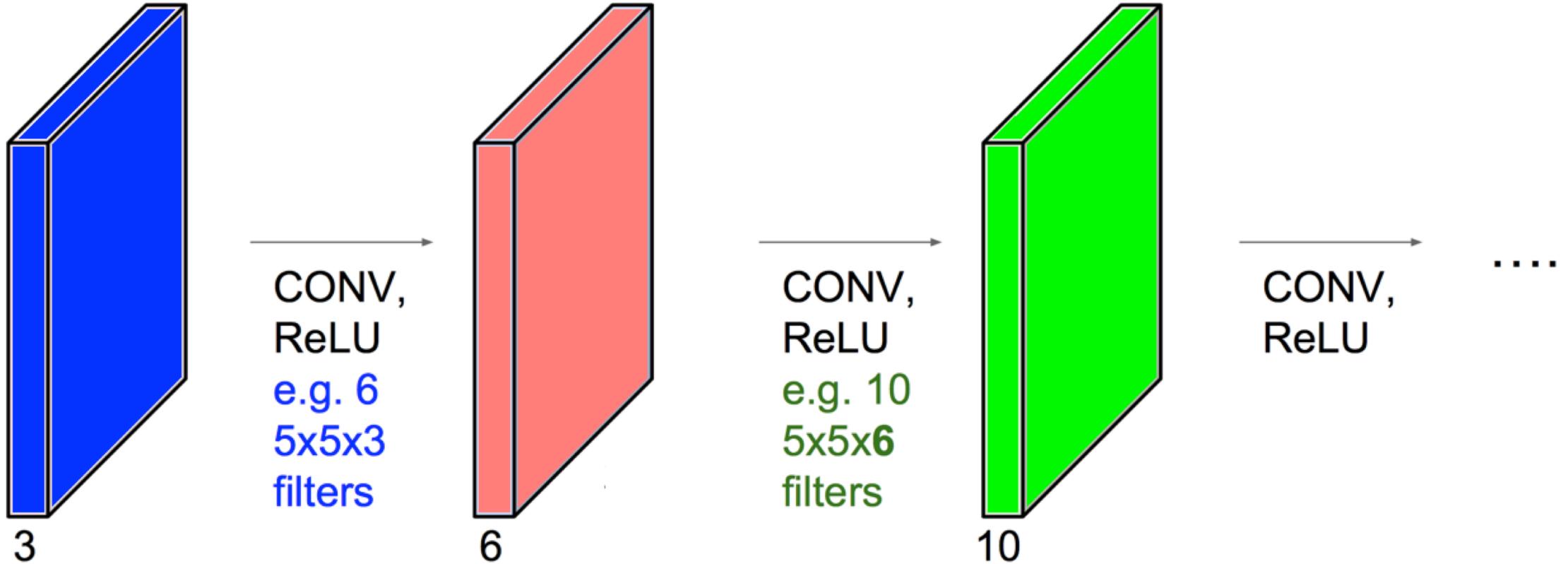
$$0 + 0 + 0 + (2 \times 1) + 0 + (1 \times -1) + 0 + 0 = 1$$

$$0 + 0 + 0 + (2 \times -1) + (1 \times 1) + 0 + (1 \times -1) + 0 + 0 = -2$$

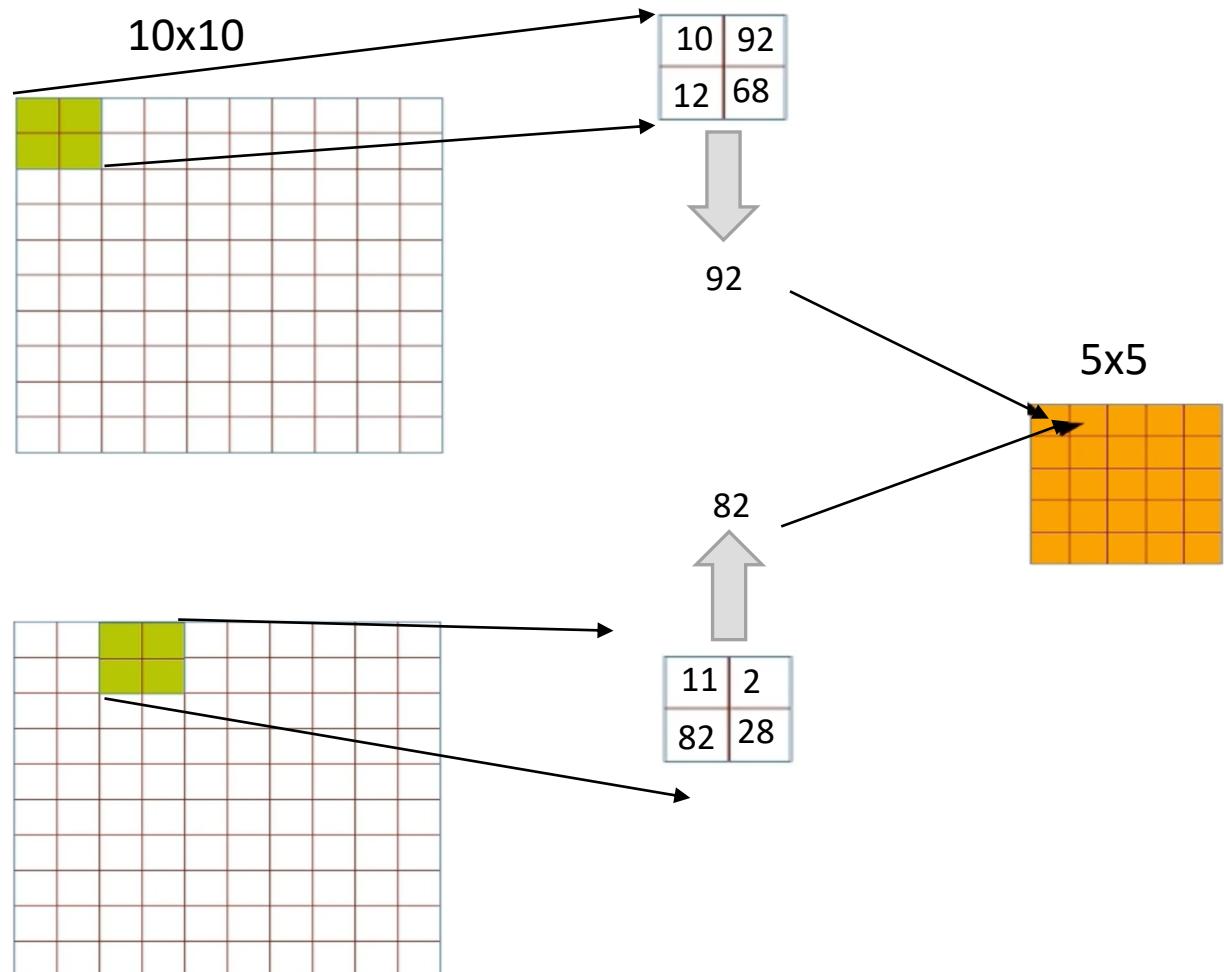
$$1 + 1 + (-2) + 9 = 9$$



Convolution Network



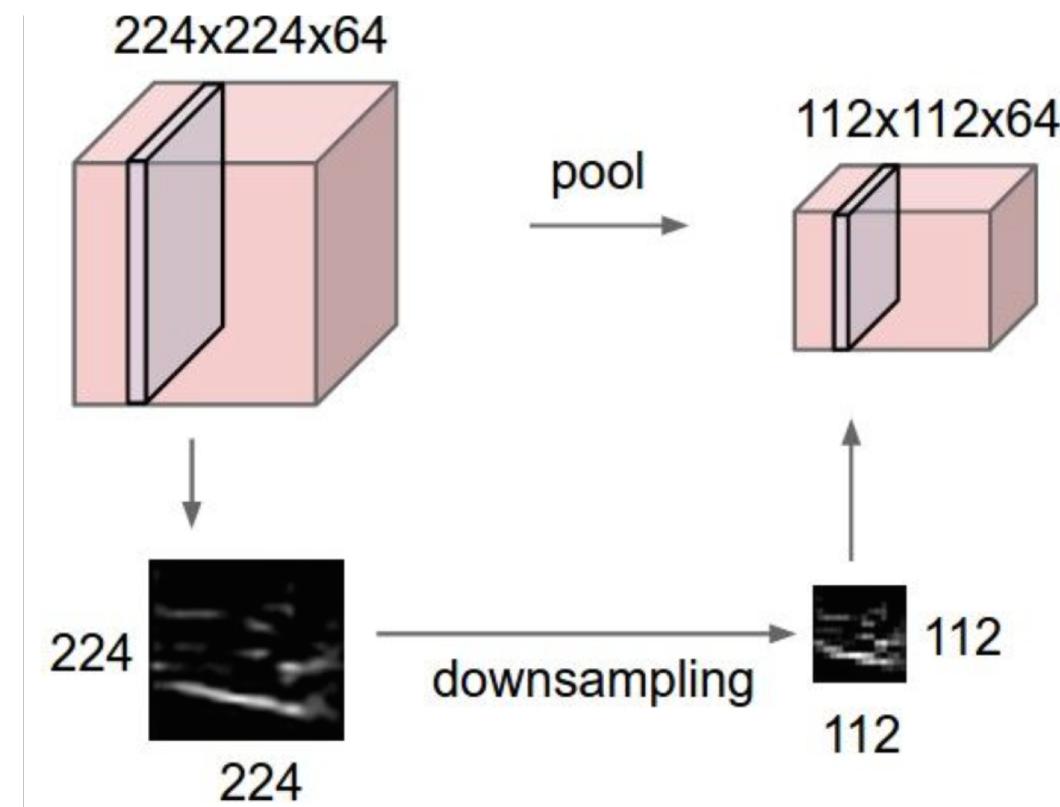
Downsampling = Max Pooling

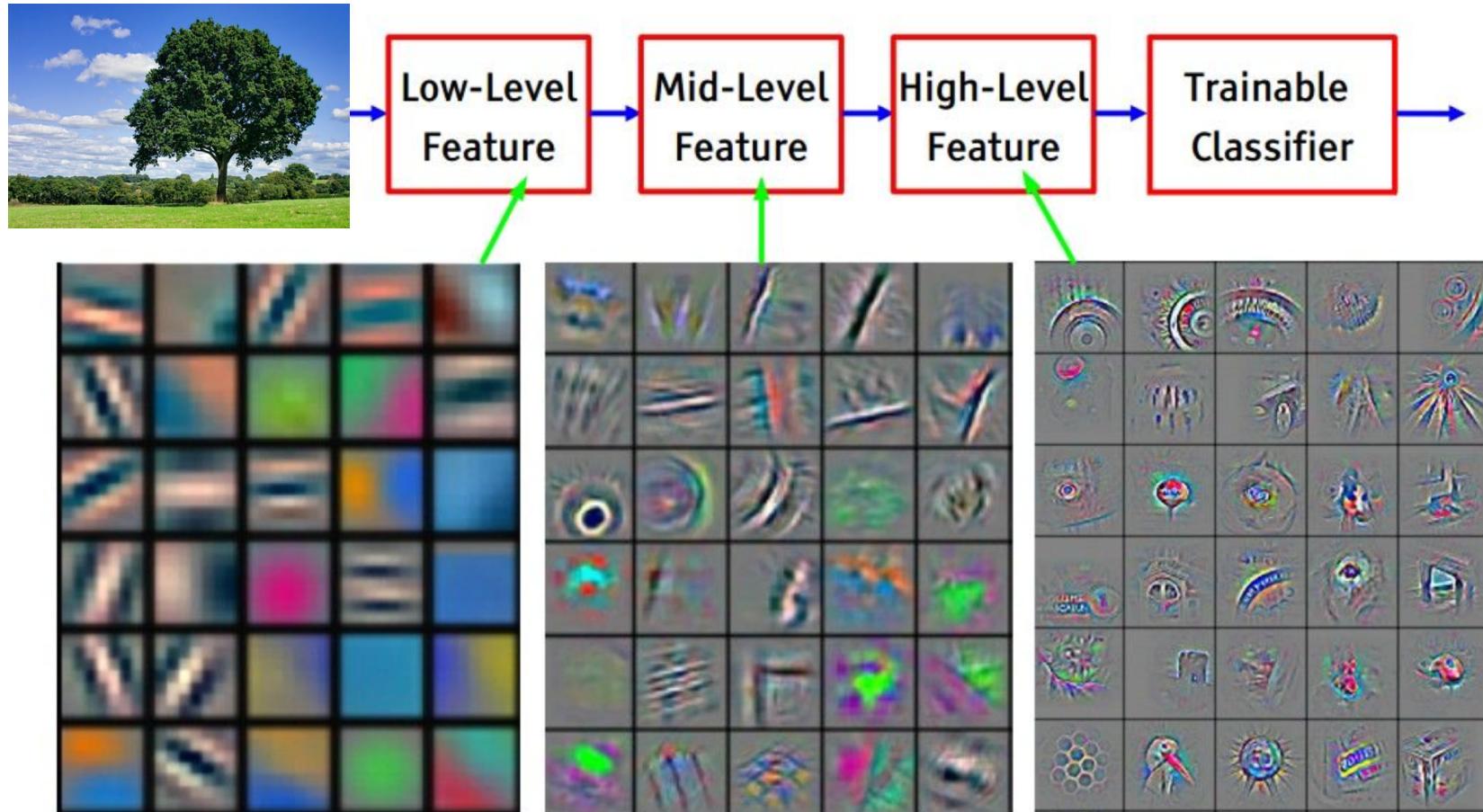


“Max Pooling” Layers to extract the “best” local feature

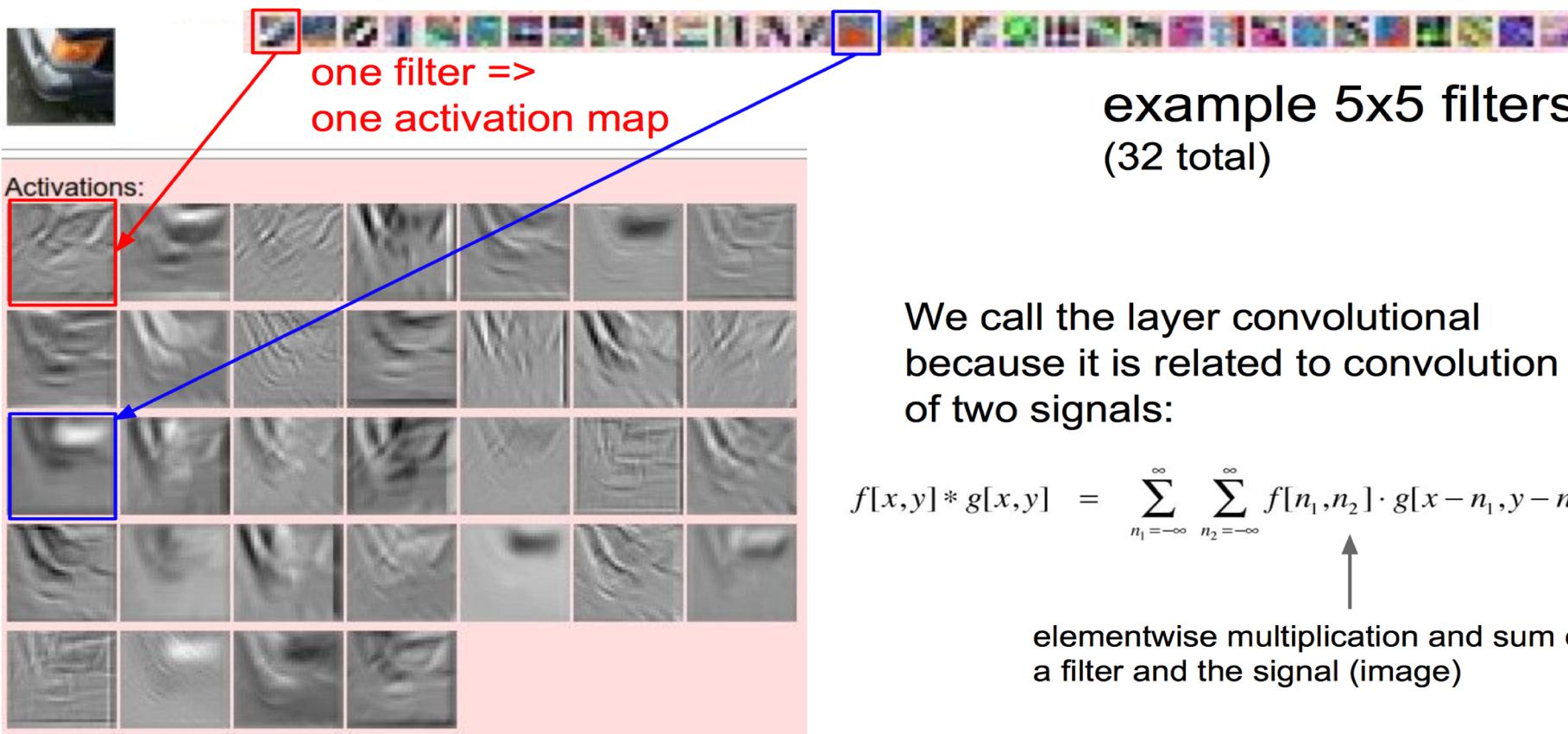
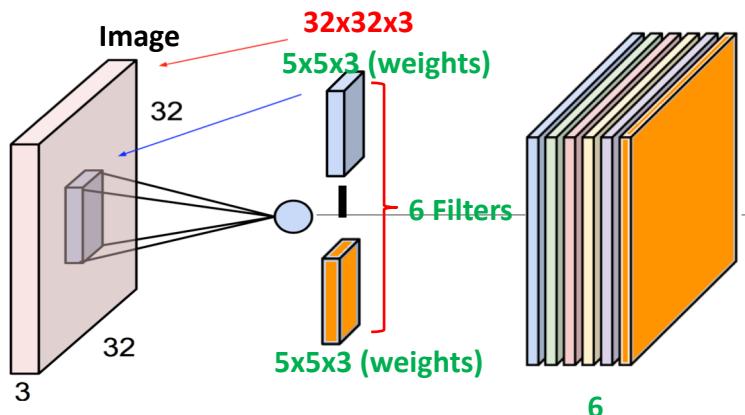
Pooling Layer

- Makes the representations smaller and more manageable
- Operates over each activation map independently





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



Architecture of LeNet-5, Convolution Neural Network

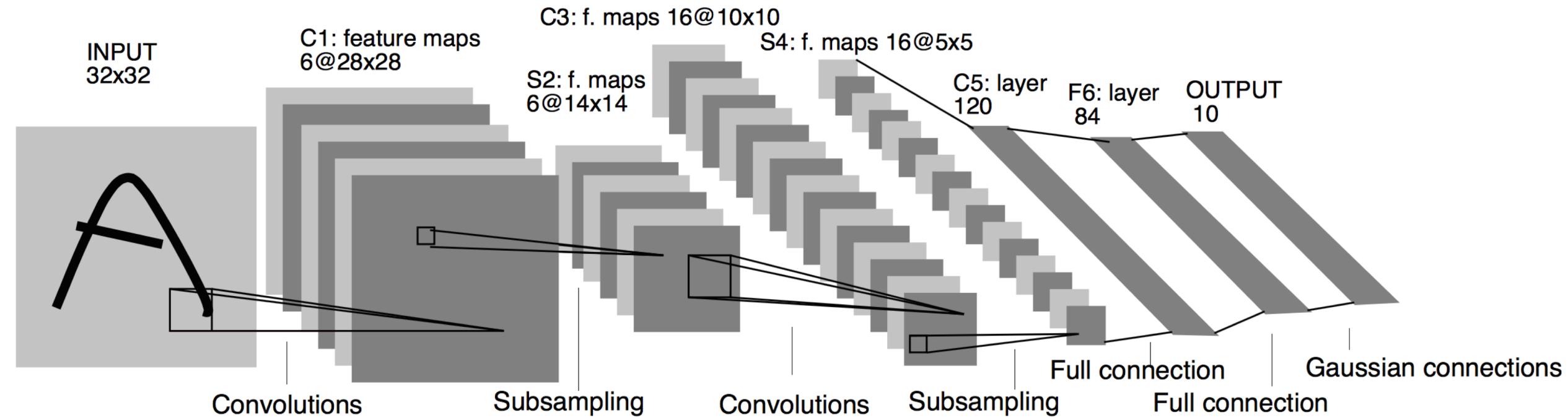
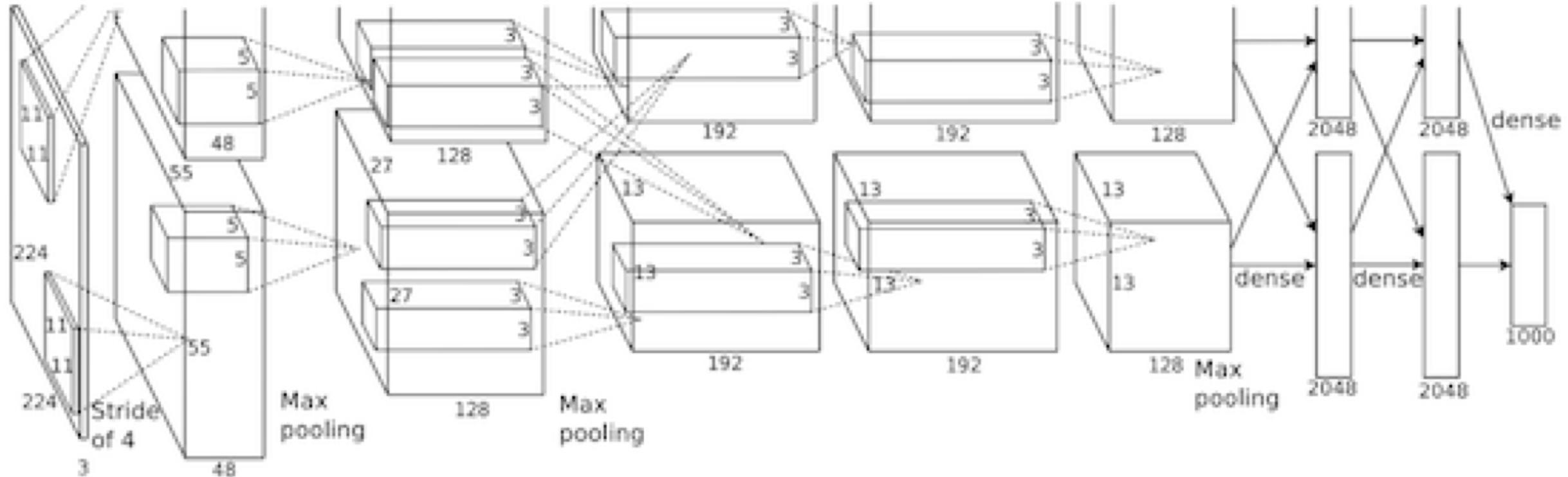


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Proc. Of the IEEE, November 1998, "Gradient-Based Learning Applied to Document Recognition"

Case Study: AlexNet



Input 227x227x3 images

First Layer (Conv1): 96 11x11 filters applied at stride 4

What is the output volume size? $(N-F)/\text{stride}+1=55$

Output volume : [55x55x96]

Parameters: $(11*11*3)*96 = 35k$

AlexNet

Full (simplified) AlexNet architecture:

[227x227x3] INPUT

[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0

[27x27x96] MAX POOL1: 3x3 filters at stride 2

[27x27x96] NORM1: Normalization layer

[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2

[13x13x256] MAX POOL2: 3x3 filters at stride 2

[13x13x256] NORM2: Normalization layer

[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1

[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1

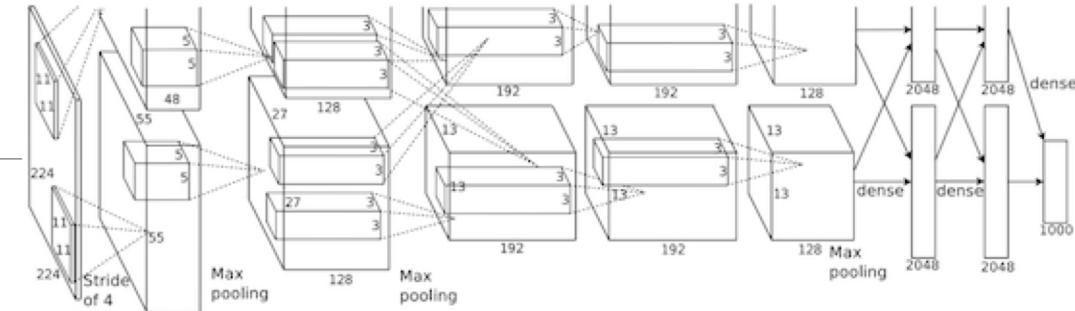
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1

[6x6x256] MAX POOL3: 3x3 filters at stride 2

[4096] FC6: 4096 neurons

[4096] FC7: 4096 neurons

[1000] FC8: 1000 neurons (class scores)



Details/Retrospectives:

- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%