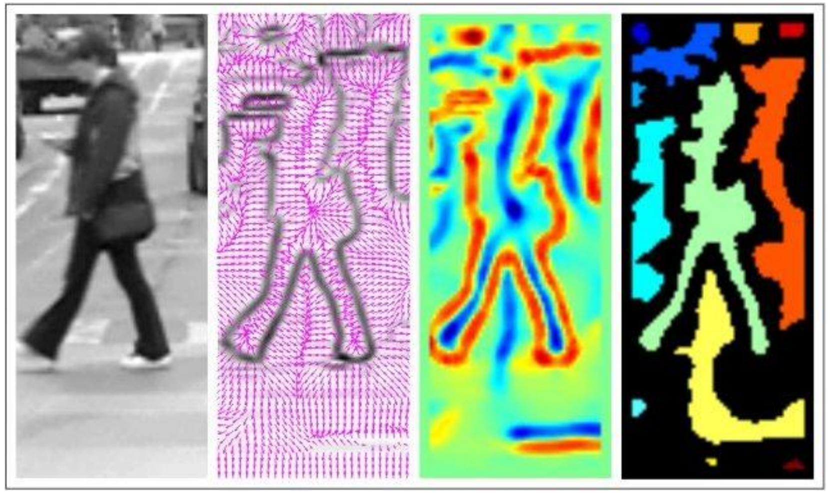
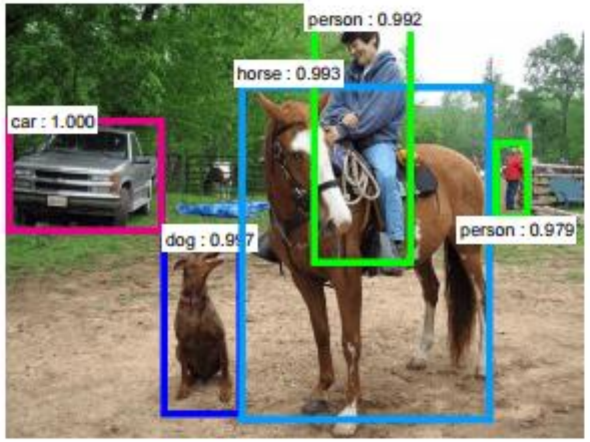
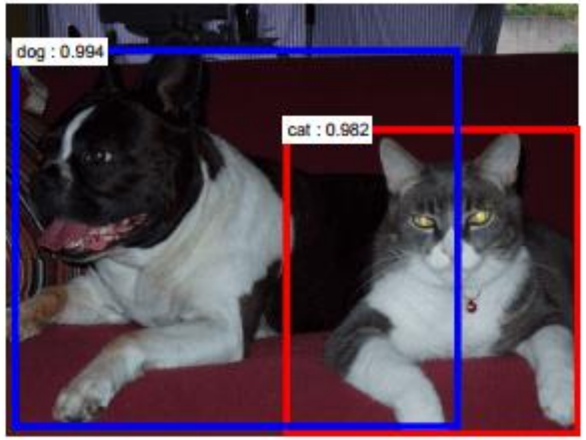
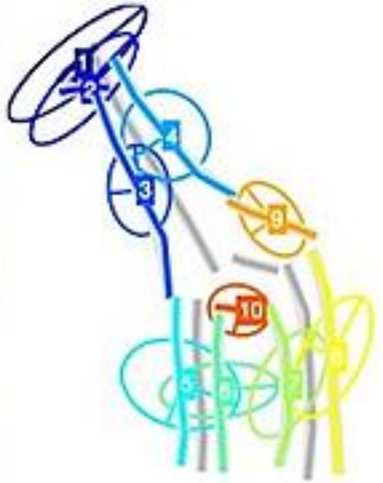
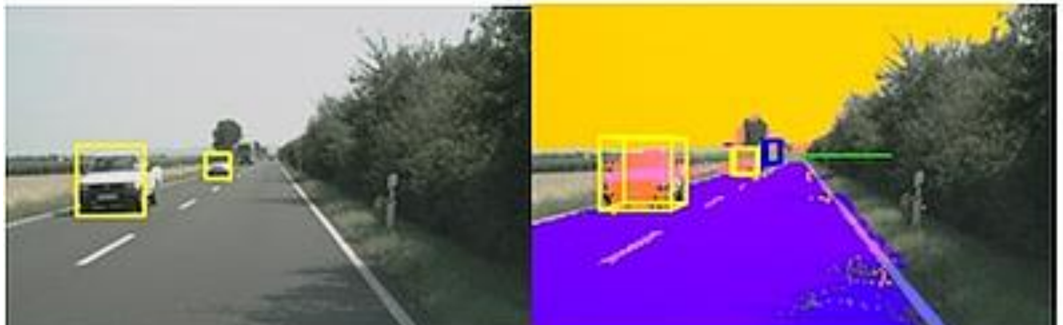
Convolutional Neural Networks

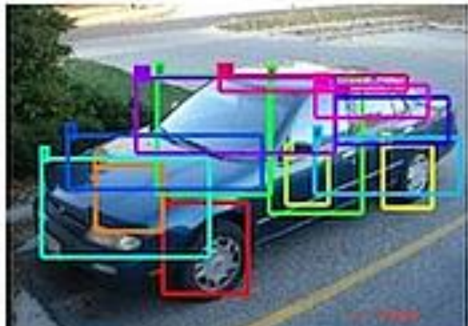



person : 0.992
horse : 0.993
car : 1.000
dog : 0.997
person : 0.979


dog : 0.994
cat : 0.982


1
2
3
4
9
10





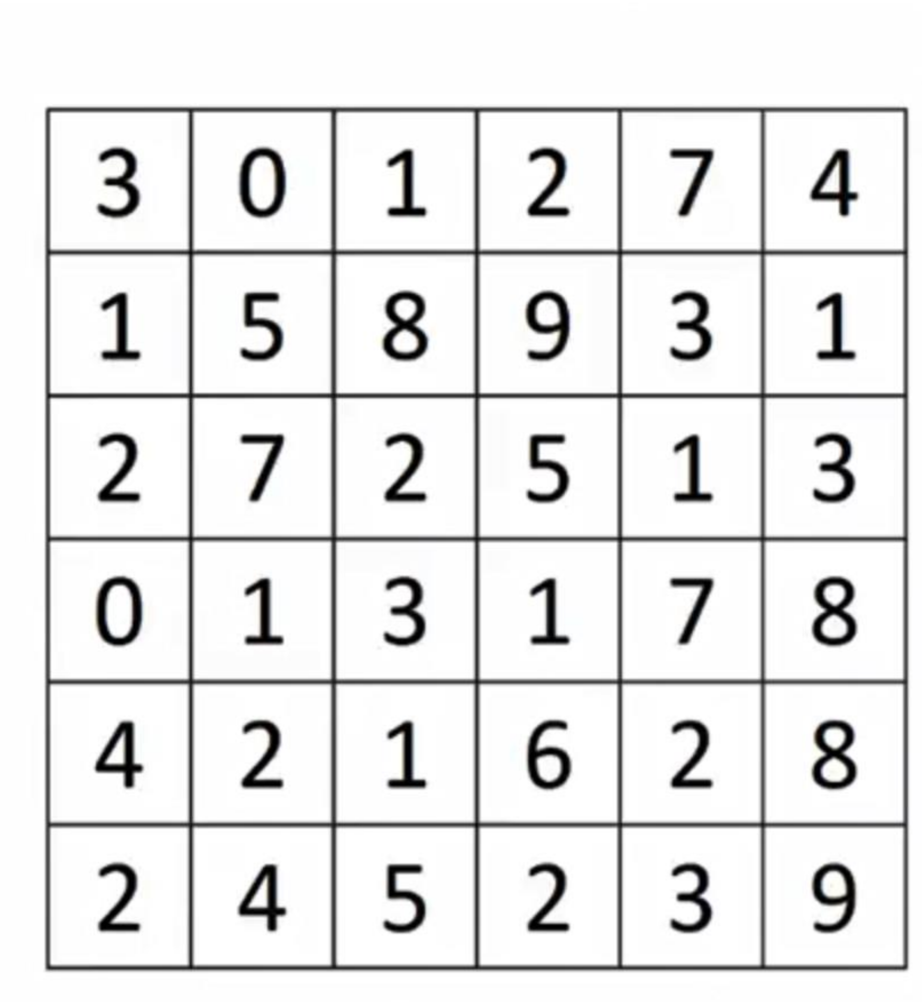
Convolution

- Convolution is a pointwise multiplication of two functions to produce a third function.
- Primary purpose of convolution in CNN is to extract features from the input image.
- matrix formed by sliding the filter over the image and computing the dot product is called the ‘Convolved Feature’ or ‘Activation Map’ or the ‘Feature Map‘.

CNNs Vs. ANNs

- ANNs suffer from curse of dimensionality when it comes to high resolution images
- CNNs do a little pre-processing, that means the network learns the filters before doing the real classification.
- We use filters (receptive fields) to exploit spatial locality by enforcing a local connectivity pattern between neurons of adjacent layers

Detecting Vertical edges



3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

6X6 Matrix (nXn)

Convolution

*

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

3X3 Filter (fXf)

=

-5	-4	0	8

(n-f+1)X(n-f+1)

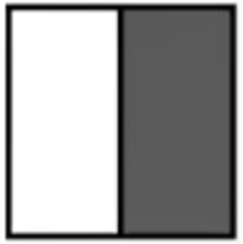
Detecting Vertical edges

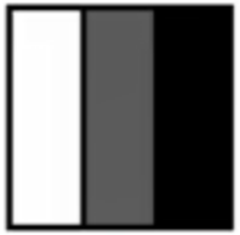
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

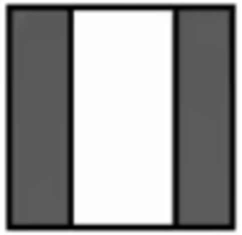
*

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

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0







Filter Weights

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

Horizontal Filter

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

Sobel Filter

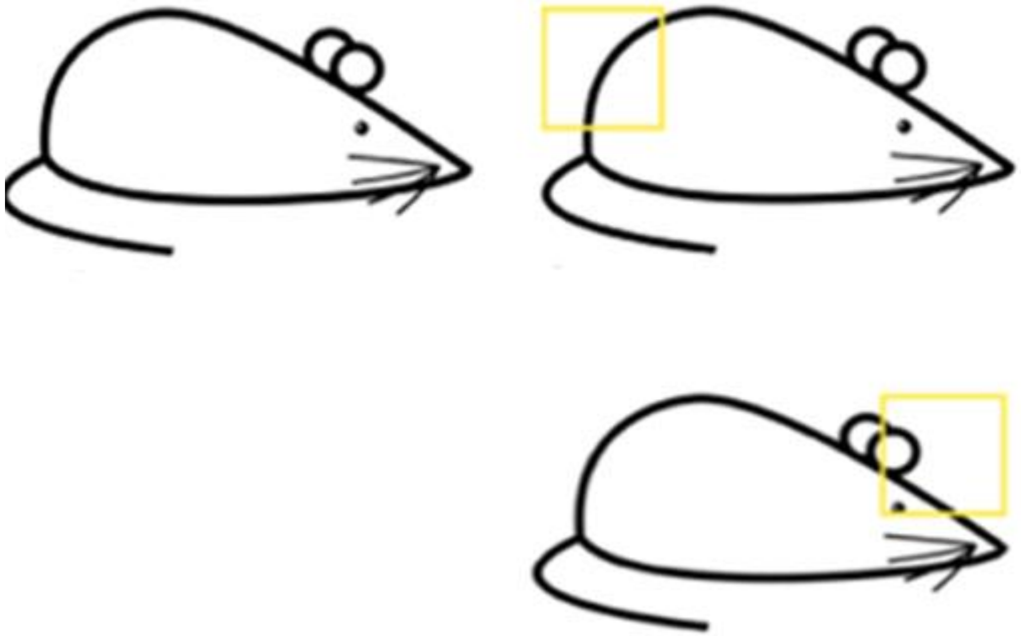
3	0	-3
10	0	-10
3	0	-3

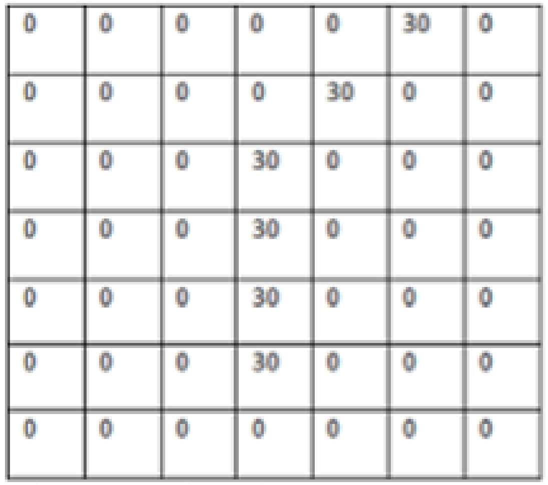
Schorr Filter

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Convolutional Neural Networks automatically estimates the weights of the filter

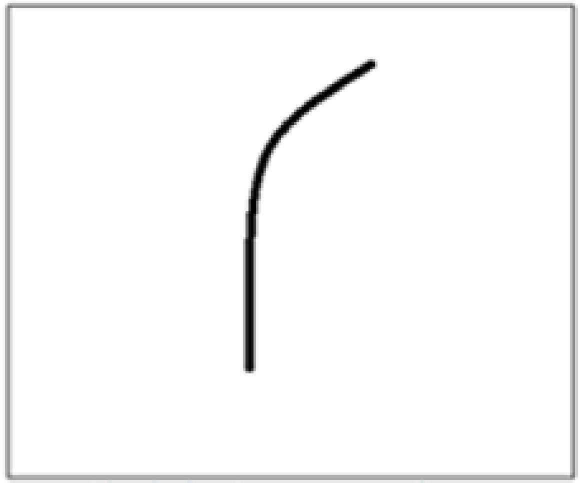
More Intuition





0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

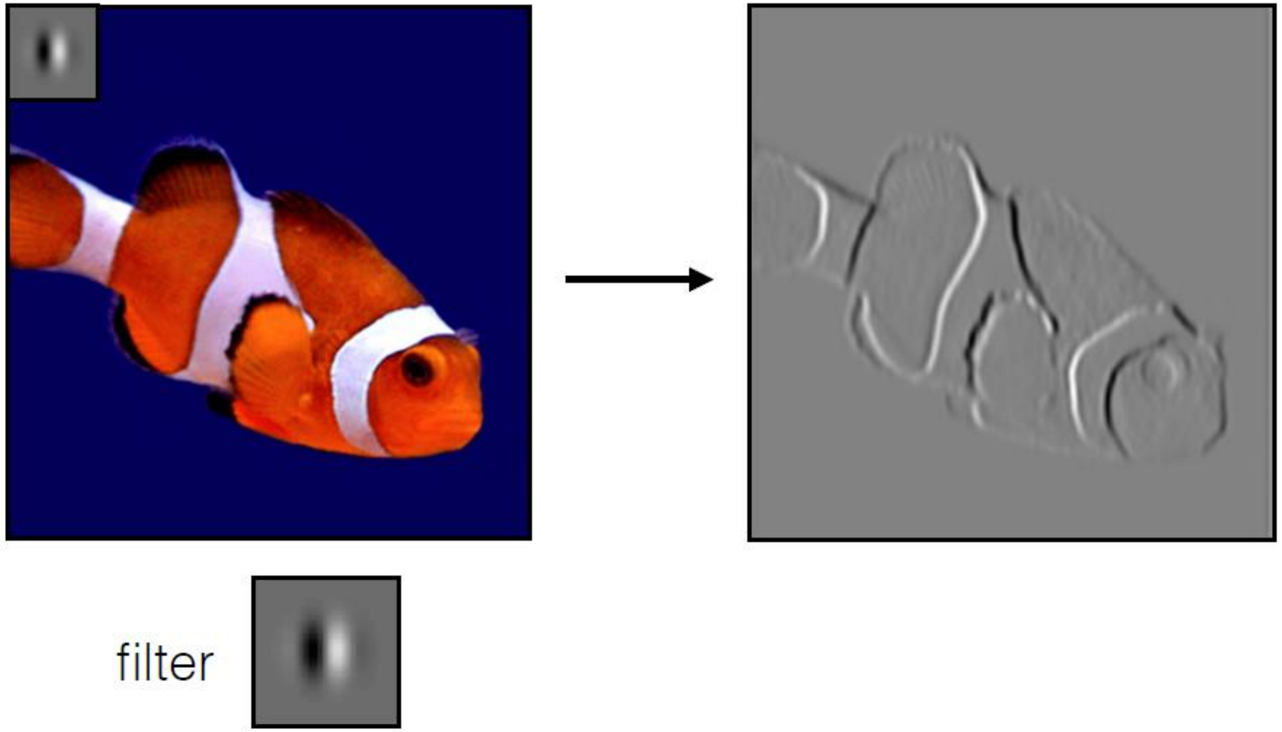


Visualization of a curve detector filter

Interpretation

Convolution is just another way of computing $W^T X$

In CNN, **input** is **image**, **kernel** is **convolution filter** to be learned, **response** is the **feature map**



Padding

Padding is used to preserve the original dimensions of the input

Zeros are added to outside of the input

Number of zero layers depend upon the size of the kernel

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	1	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

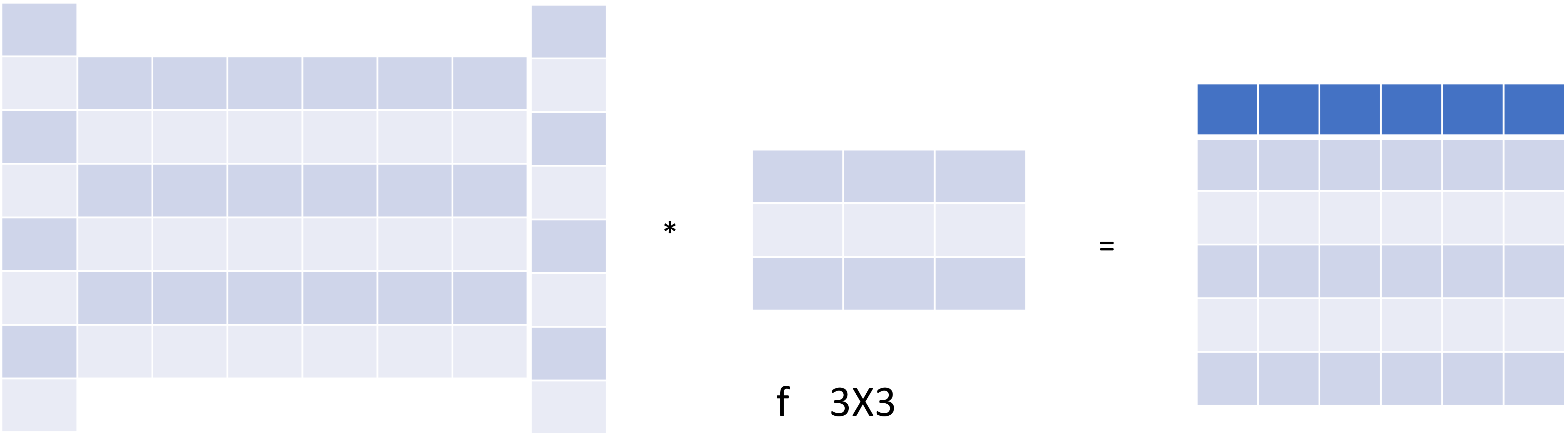
5X5 (with padding)

×1	×0	×1
×0	×1	×0
×1	×0	×1

2	2	3	1	1
1	4	3	4	1
2	2	4	3	3
1	2	3	4	1
1	2	3	1	1

5X5

Padding



f 3X3

nXn 6X6 to 8X8 Padding=1

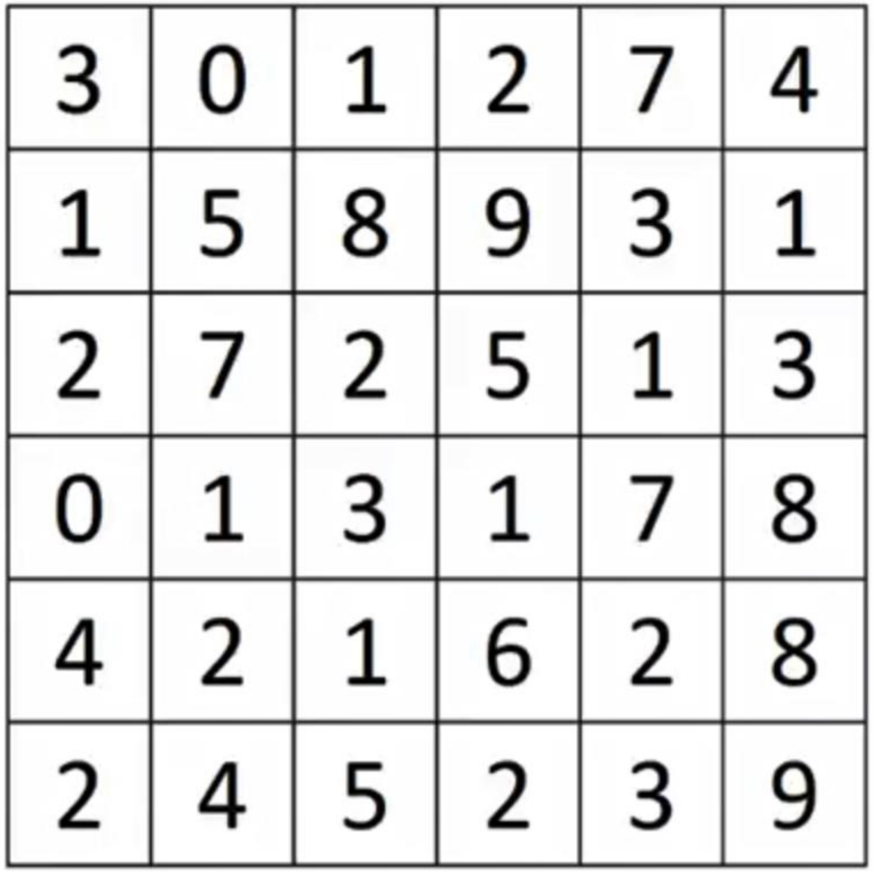
(n-f+1)X(n-f+1) to (n+2p-f+1)X(n+2p-f+1)

Valid to same

Stride=s

$$\text{Floor}\left(\frac{n+2p-f}{s}+1\right) X\text{Floor}\left(\frac{n+2p-f}{s}+1\right)$$

Stride



3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

6X6 Matrix

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

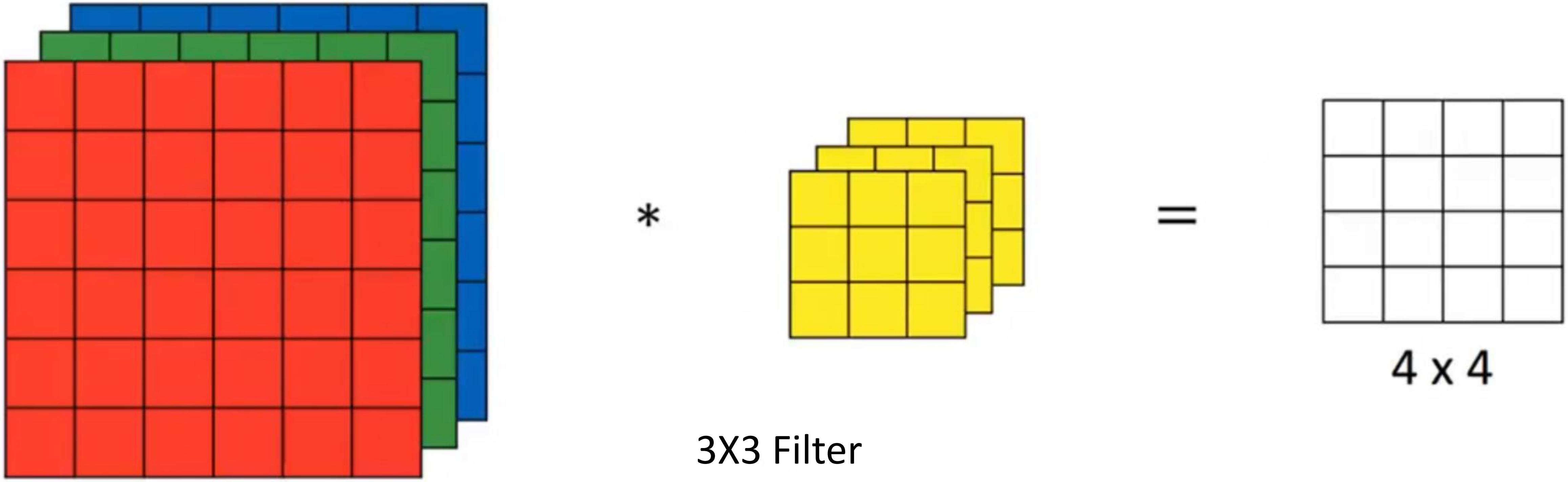
3X3 Filter

-5	8
-3	-16

2X2

Stride=s (3 Here) Floor($\frac{n+2p-f}{s} + 1$) XFloor($\frac{n+2p-f}{s} + 1$)

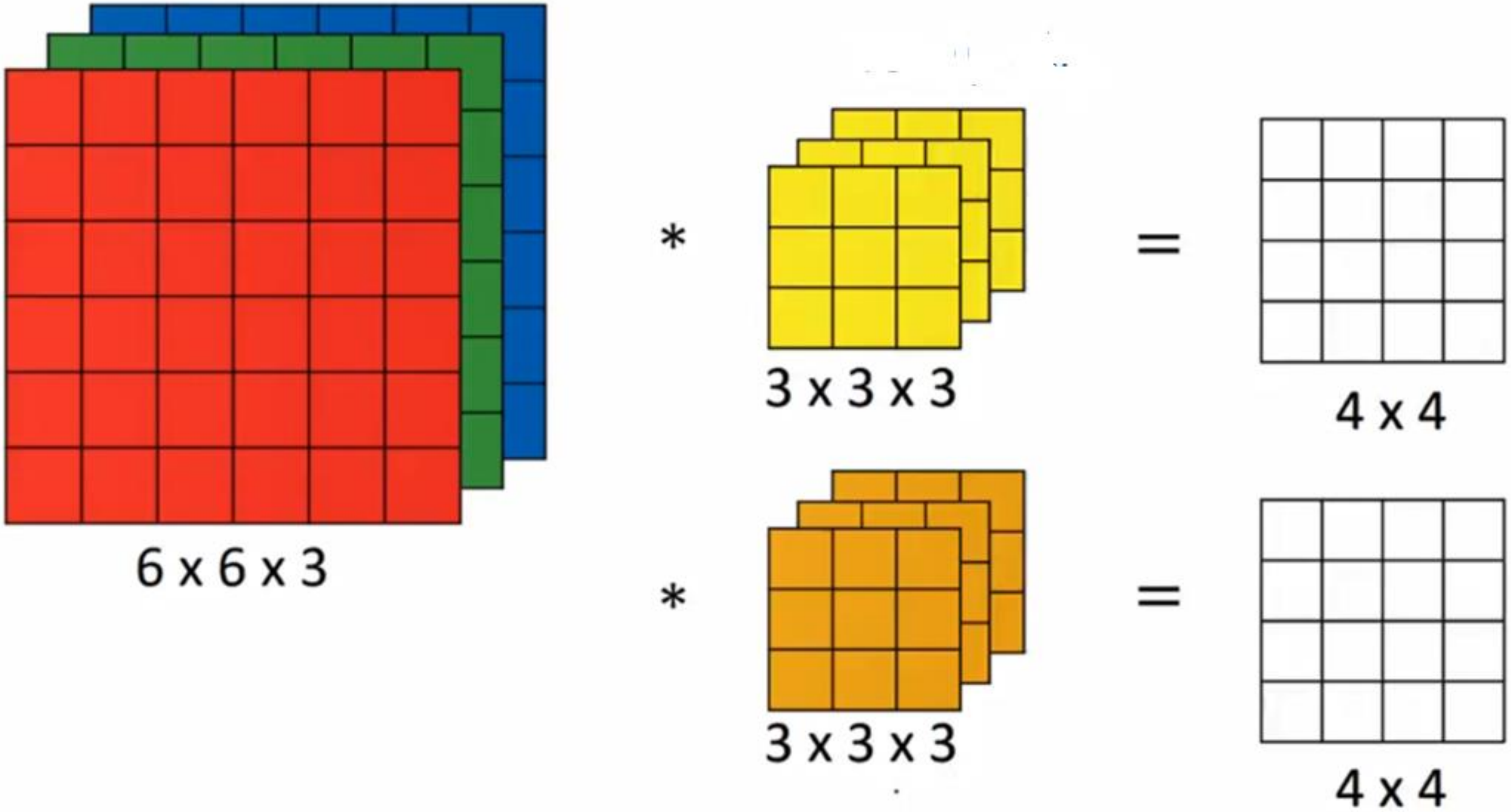
Channels



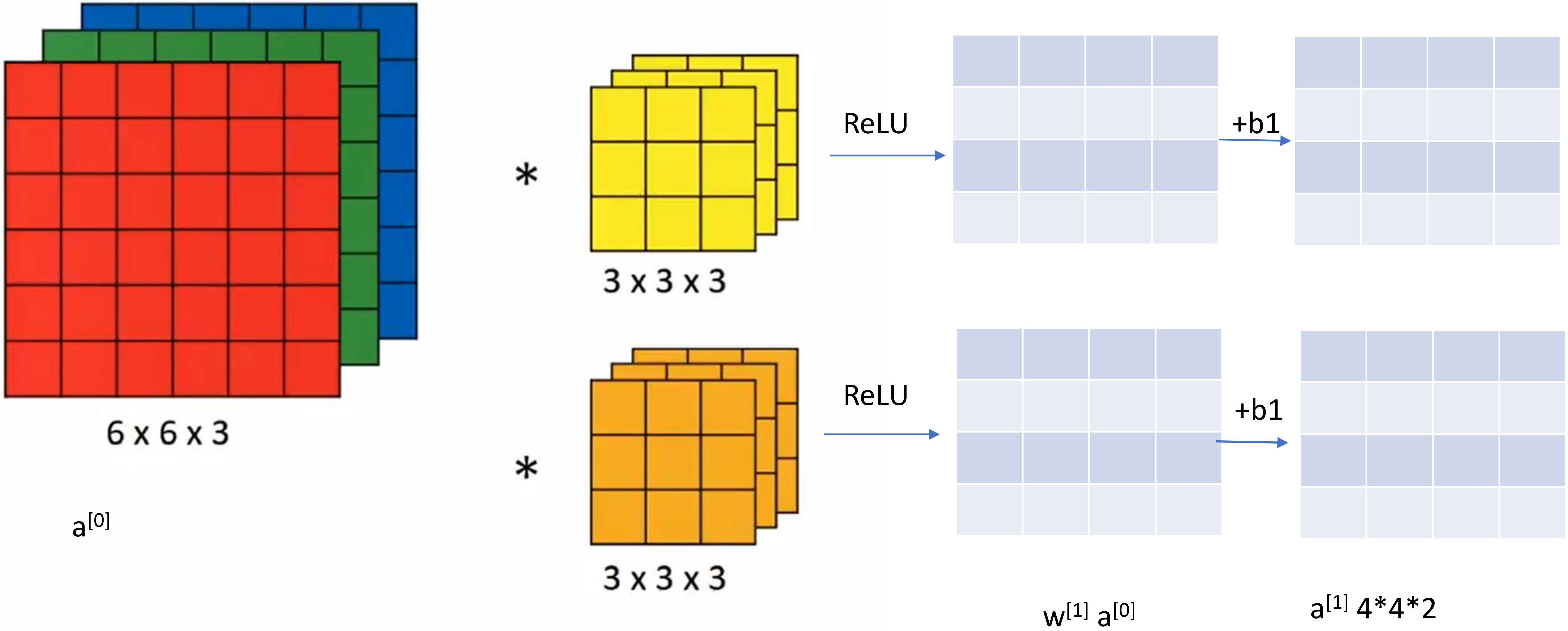
3X3 Filter

6X6 Matrix

Padding =0 Stride=1



$nXnXn_c$ $fXfXn_c$ $n\text{-}f\text{+}1Xn\text{-}f\text{+}1Xn_{c'}$ c'=no of filters



6 x 6 x 3

$a^{[0]}$

*

3 x 3 x 3

ReLU

+b1

*

3 x 3 x 3

ReLU

+b1

$w^{[1]}$ $a^{[0]}$

$a^{[1]}$ 4*4*2

$w^{[1]}$ 2 filters means two units here

Pooling

1	4	6	3
1	8	9	7
2	9	1	2
3	4	4	3

Max Pooling : One example of pooling layer

8	9
9	4

f=2

s=2   4X4 converted to 2X2

Function of Pooling is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network.

Pooling

1	4	6	3
1	8	9	7
2	9	1	2
3	4	4	3

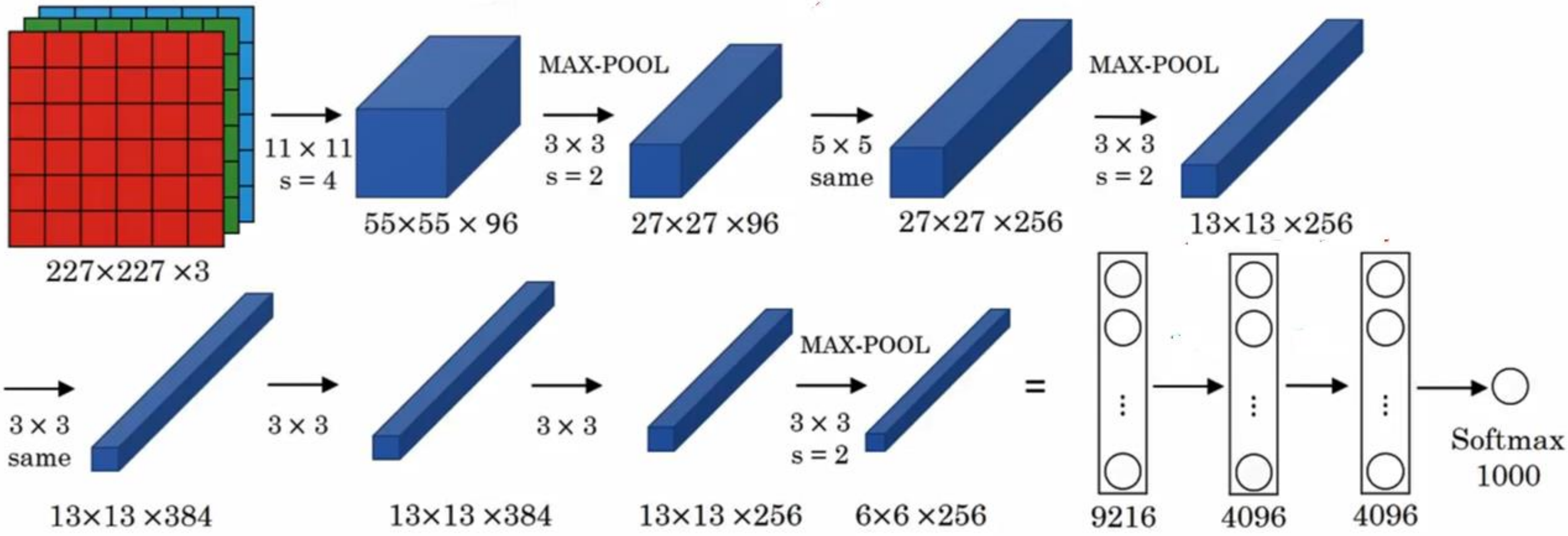
3.5	6.25
4.5	2.5

Average Pooling : Another example of pooling layer

f=2

s=2 4X4 converted to 2X2

AlexNet


227×227 ×3
11 × 11
s = 4
55×55 × 96
MAX-POOL
3 × 3
s = 2
27×27 ×96
5 × 5
same
27×27 ×256
MAX-POOL
3 × 3
s = 2
13×13 ×256
3 × 3
same
13×13 ×384
3 × 3
13×13 ×384
3 × 3
13×13 ×256
MAX-POOL
3 × 3
s = 2
6×6 ×256
=
9216
4096
4096
Softmax
1000

Famous CNN Models

- LeNet - 1990
- AlexNet - 2012
- ZFNet - 2013
- GoogLeNet - 2014
- VGGNet - 2014
- ResNet - 2015
- Inception v3 - 2016
- MobileNet - 2017