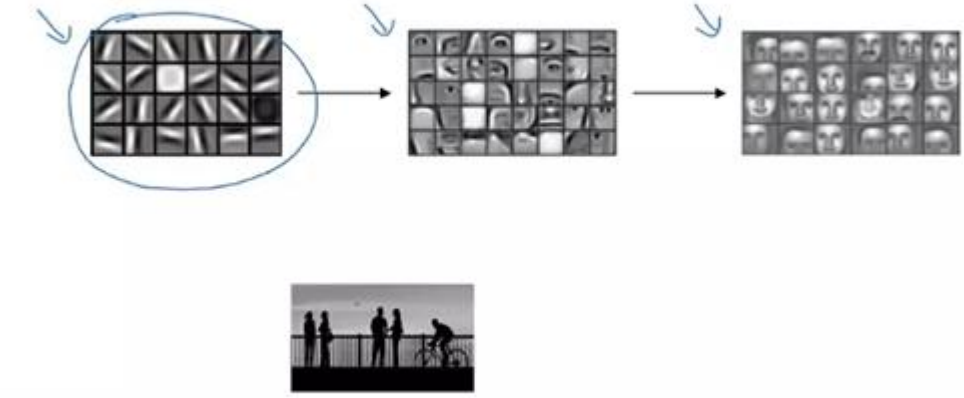
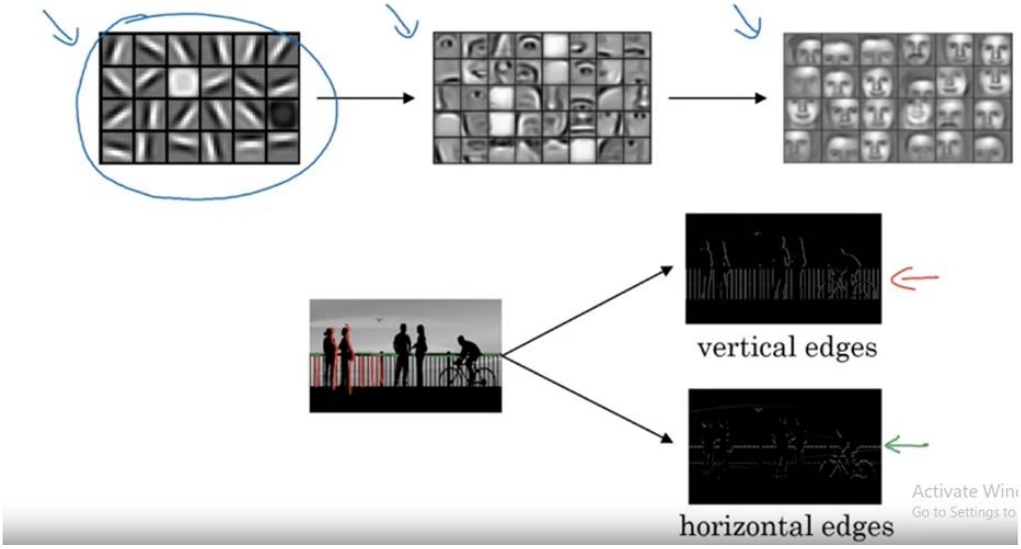
Computer Vision Problem



Computer Vision Problem

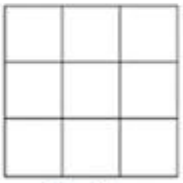


Vertical edge detection

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

6×6

*



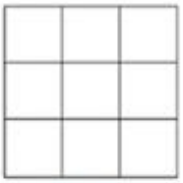
3×3 filter

Vertical edge detection

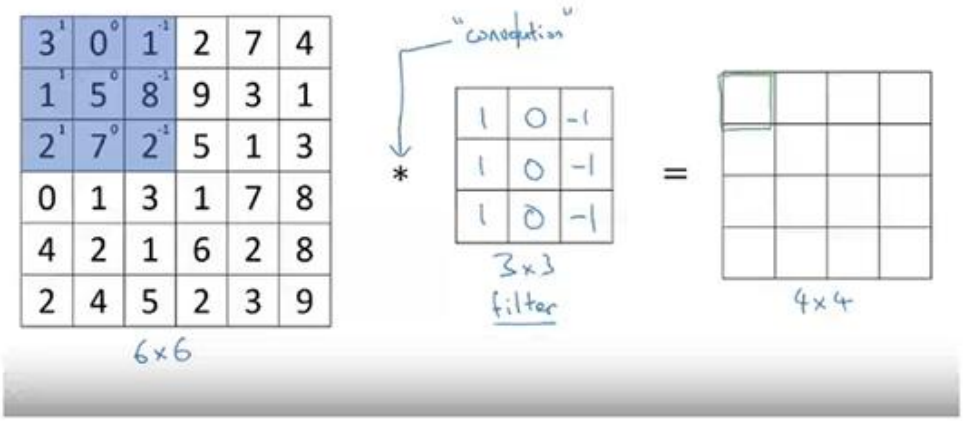
3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

6×6

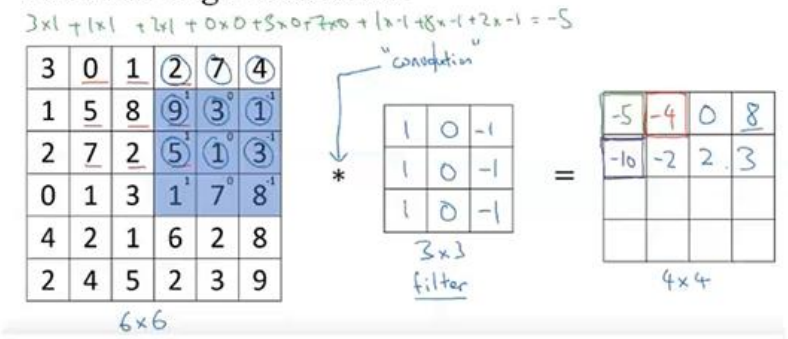
*



3×3 filter

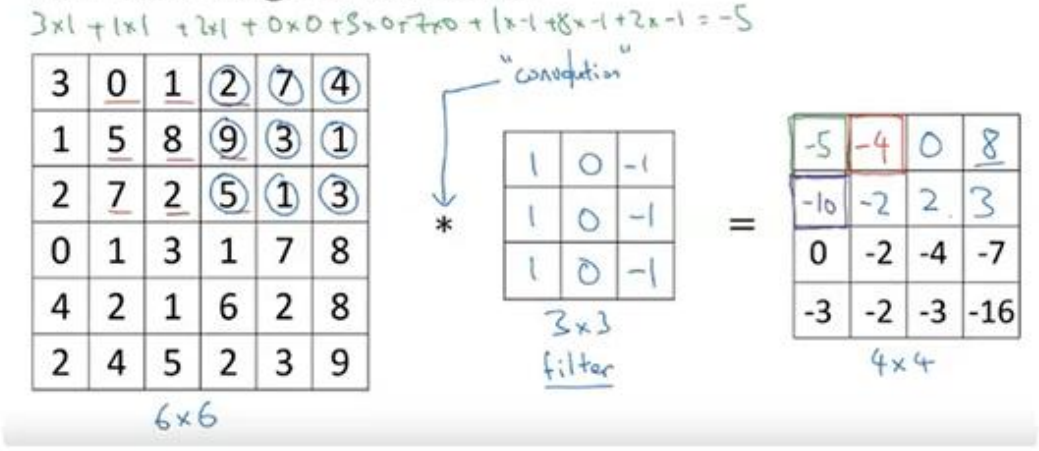


Vertical edge detection



Fill remaining cells

Vertical edge detection



Vertical edge detection

$3\times1 + 1\times1 + 2\times1 + 0\times0 + 5\times0 + 7\times0 + 1\times-1 + 8\times-1 + 2\times-1 = -5$

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

6×6

"convolution"

*

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

3×3 filter

=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

4×4

python: conv_forward
tensorflow: tf.nn.conv2d
keras: Conv2D

Vertical edge detection

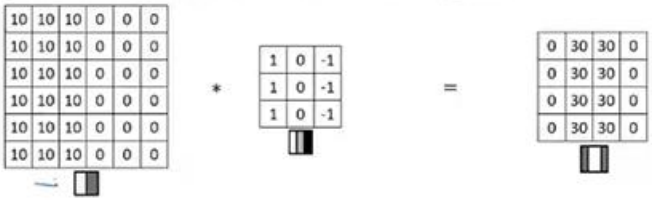
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

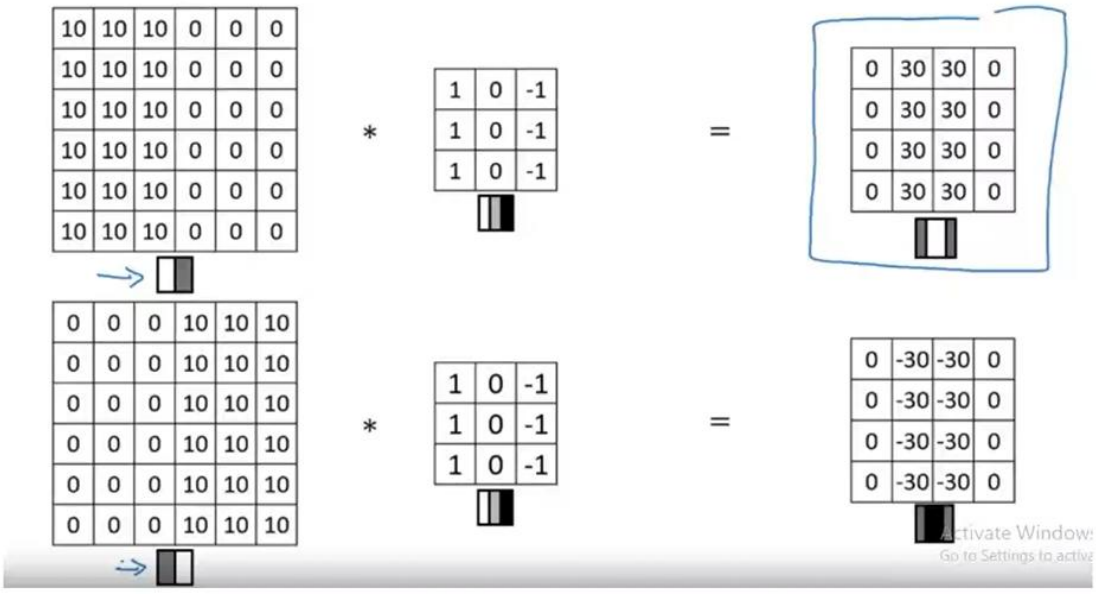
6×6

*

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

Vertical edge detection examples





Vertical and Horizontal Edge Detection

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

Vertical

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

Horizontal

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

*

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

=

0	0	0	0
30	10	-10	-30
30	10	-10	-30
0	0	0	0

Learning to detect edges

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

→

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

Sobel filter

Learning to detect edges

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

→

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

Sobel filter

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

Scharr filter

Learning to detect edges

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

→

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

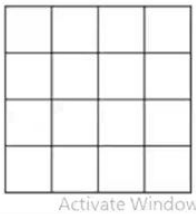
Sobel filter

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

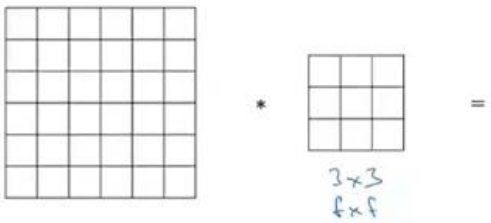
Scharr filter

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9



Padding

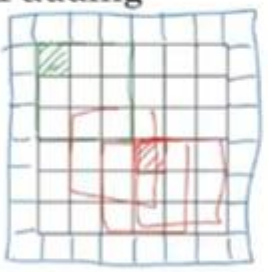


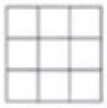
6×6
n×n

n-f+1 × n-f+1 → 4×4
6-

Padding

- shrinky output
- throw away info from edge



* 

3×3
f×f

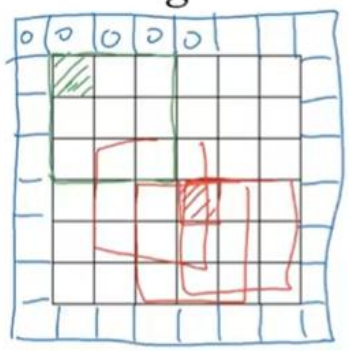
=

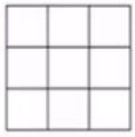
6×6 → 8×8
n×n

n-f+1 × n-f+1
6-3+1=4 → 4×4

Padding

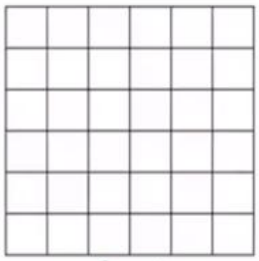
- shrinky output
- throw away info from edge



* 

3×3
f×f

=



6×6

6×6 → 8×8
n×n

n-f+1 × n-f+1
6-3+1=4 → ~~4×4~~

p = padding = 1

n+2p-f+1 × n+2p-f+1
6+2-3+1 × ___ = 6×6

Valid and Same convolutions

"Valid": n×n * f×f → n-f+1 × n-f+1
6×6 * 3×3 → 4×4

"Same": Pad so that output size is the same as the input size.

n+2p-f+1 × n+2p-f+1

Valid and Same convolutions

↳ no padding

"Valid": $n \times n \quad * \quad f \times f \quad \rightarrow \quad n-f+1 \times n$

$6 \times 6 \quad * \quad 3 \times 3 \quad \rightarrow \quad 4 \times 4$

"Same": Pad so that output size is the same as the input size.

f is usually odd

$n+2p-f+1 \times n+2p-f+1$

Valid and Same convolutions

↳ no padding

"Valid": $n \times n \quad * \quad f \times f \quad \rightarrow \quad n-f+1 \times n-f+1$

$6 \times 6 \quad * \quad 3 \times 3 \quad \rightarrow \quad 4 \times 4$

"Same": Pad so that output size is the same as the input size.

f is usually odd

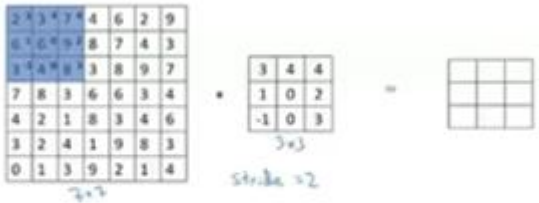
$n+2p-f+1 \times n+2p-f+1$

$n+2p-f+1 = n \Rightarrow p = \frac{f-1}{2}$

$3 \times 3 \quad p = \frac{3-1}{2} = 1 \quad | \quad 5 \times 5 \quad p = 2$

Click to add te

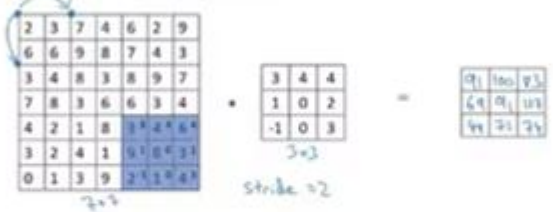
Strided convolution



Click to add title

• Click to add te

Strided convolution



Strided convolution

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3^3	4^4	6^4
3	2	4	1	9^1	8^0	3^2
0	1	3	9	2^{-1}	1^0	4^3

7×7

*

3	4	4
1	0	2
-1	0	3

3×3

Stride = 2

=

91	100	83
69	91	127
44	72	74

3×3

n×n * f×f

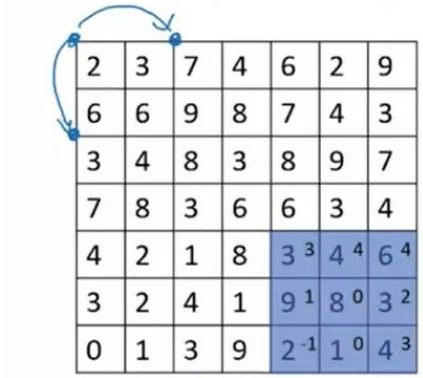
padding p stride s

s=2

$\frac{n+2p-f}{s}+1 \times \frac{n+2p-f}{s}+1$

$\frac{7+0-3}{2}+1 = \frac{4}{2}+1 = 3$

Strided convolution



2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3^{3}	4^{4}	6^{4}
3	2	4	1	9^{1}	8^{0}	3^{2}
0	1	3	9	2^{-1}	1^{0}	4^{3}

7×7

*

3	4	4
1	0	2
-1	0	3

3×3

stride = 2

=

91	100	83
69	91	127
44	72	74

3×3

n×n * f×f

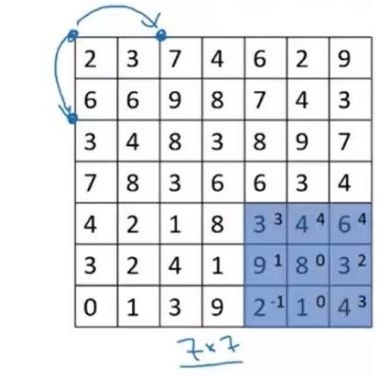
padd p stride s

s = 2

$\frac{n+2p-f}{s}+1 \times \frac{n+2p-f}{s}+1$

$\frac{7+0-3}{2}+1 = \frac{4}{2}+1 = 3$

Strided convolution



2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3^{3}	4^{4}	6^{4}
3	2	4	1	9^{1}	8^{0}	3^{2}
0	1	3	9	2^{-1}	1^{0}	4^{3}

7×7

*

3	4	4
1	0	2
-1	0	3

3×3

stride = 2

=

91	100	83
69	91	127
44	72	74

3×3

$\lfloor z \rfloor = \text{floor}(z)$

n×n * f×f

padd p stride s

s = 2

$\left\lfloor \frac{n+2p-f}{s}+1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s}+1 \right\rfloor$

$\frac{7+0-3}{2}+1 = \frac{4}{2}+1 = 3$

Technical note on cross-correlation vs. convolution

Convolution in math textbook:

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

*

3	4	5
1	0	2
-1	9	7

Technical note on cross-correlation vs. convolution

Convolution in math textbook:

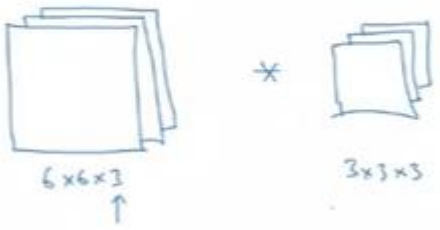
2	3	7	4	6	2
6	6	9	8	7	4
3	4	8	3	8	9
7	8	3	6	6	3
4	2	1	8	3	4
3	2	4	1	9	8

*

3	4	5
1	0	2
-1	9	7

7	9	-1
2	0	1
5	4	3

Convolutions on RGB images



Technical note on cross-correlation vs. convolution

Convolution in math textbook:

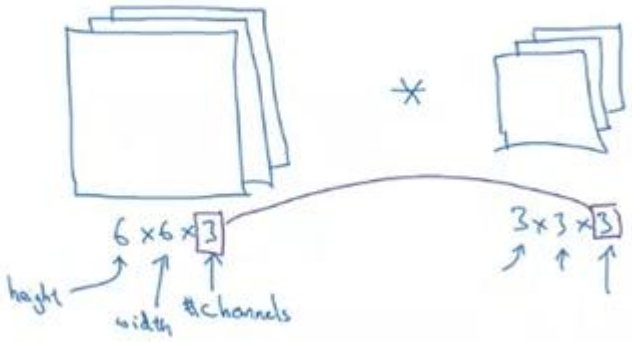
2	3	7	4	6	2
6	6	9	8	7	4
3	4	8	3	8	9
7	8	3	6	6	3
4	2	1	8	3	4
3	2	4	1	9	8

*

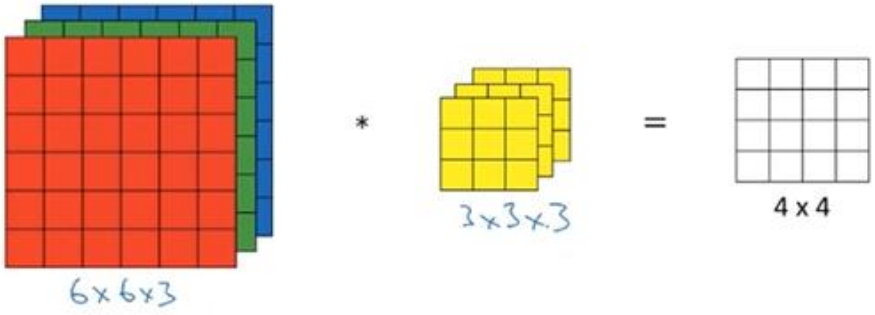
3	4	5
1	0	2
-1	9	7

7	9	-1
2	0	1
5	4	3

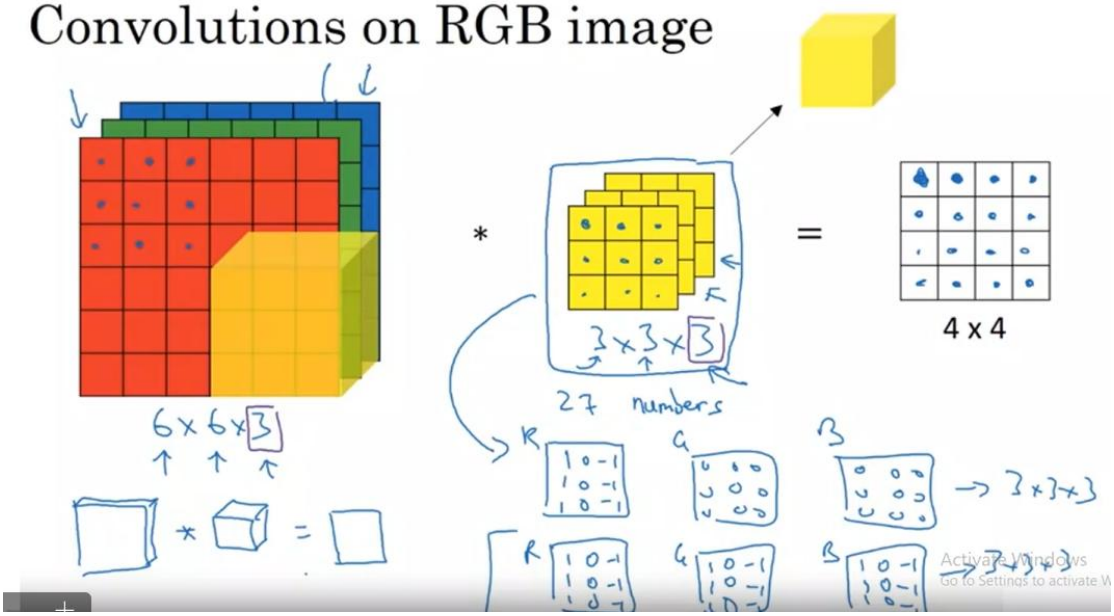
Convolutions on RGB images



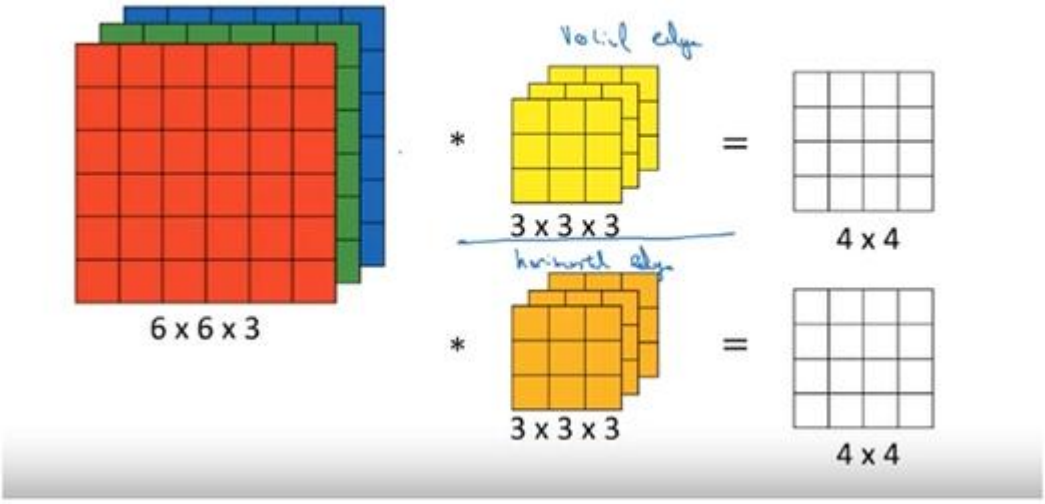
Convolutions on RGB image



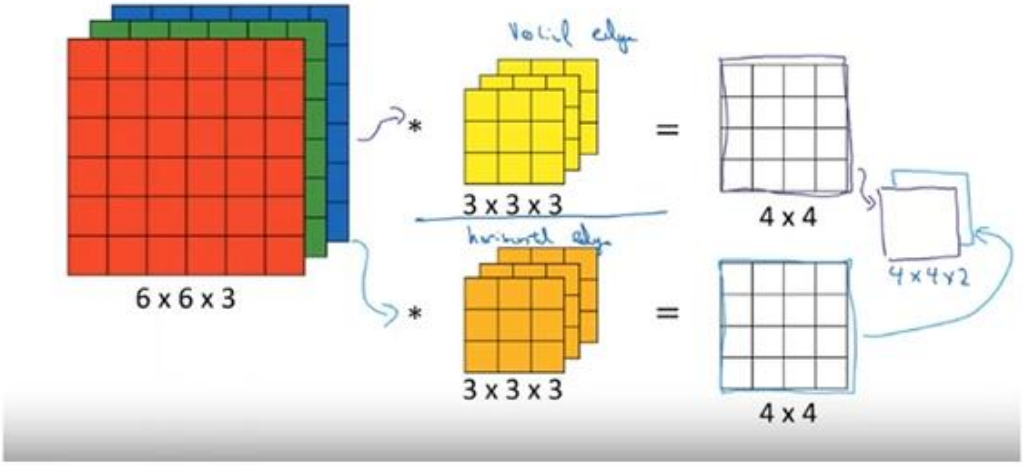
Convolutions on RGB image



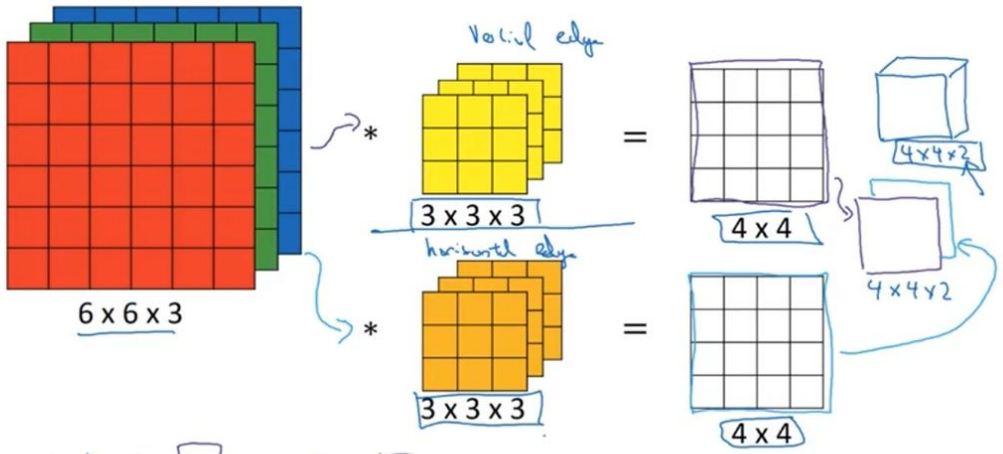
Multiple filters



Multiple filters

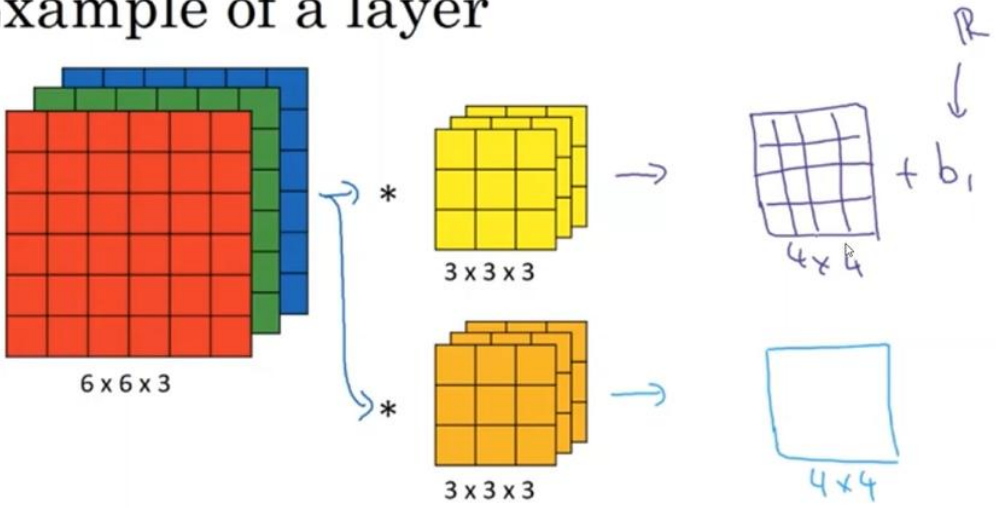


Multiple filters

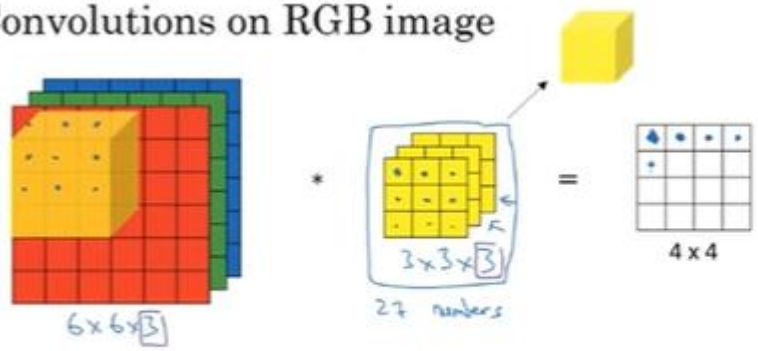


Summary: $n \times n \times n_c$ $6 \times 6 \times 3$ * $f \times f \times n_c$ $3 \times 3 \times 3$ →

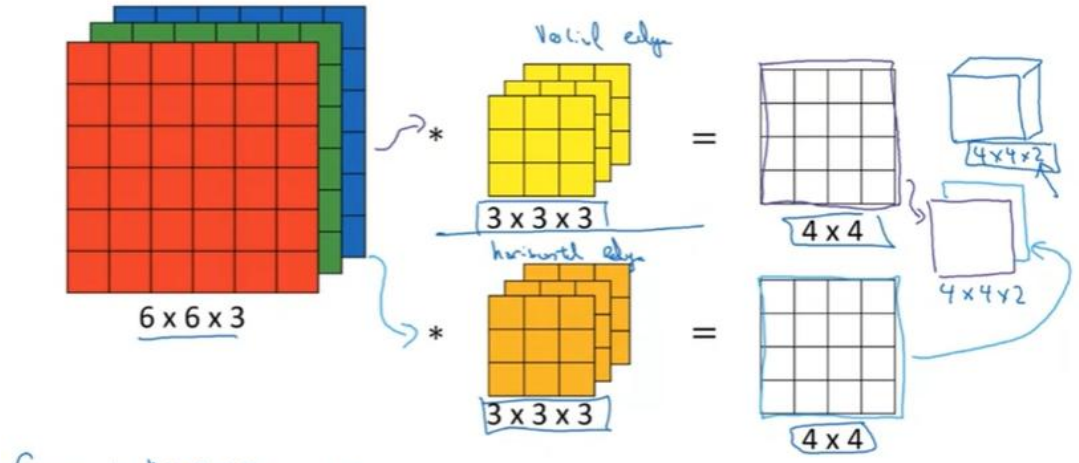
Example of a layer



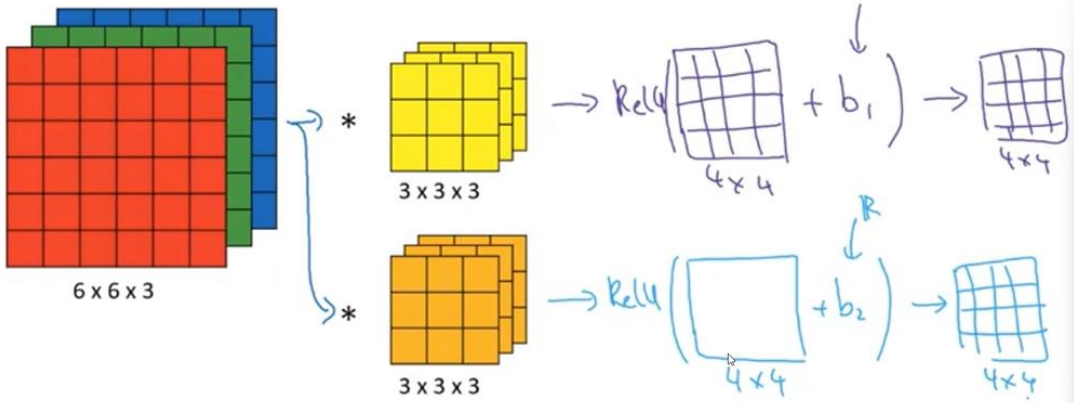
Convolutions on RGB image



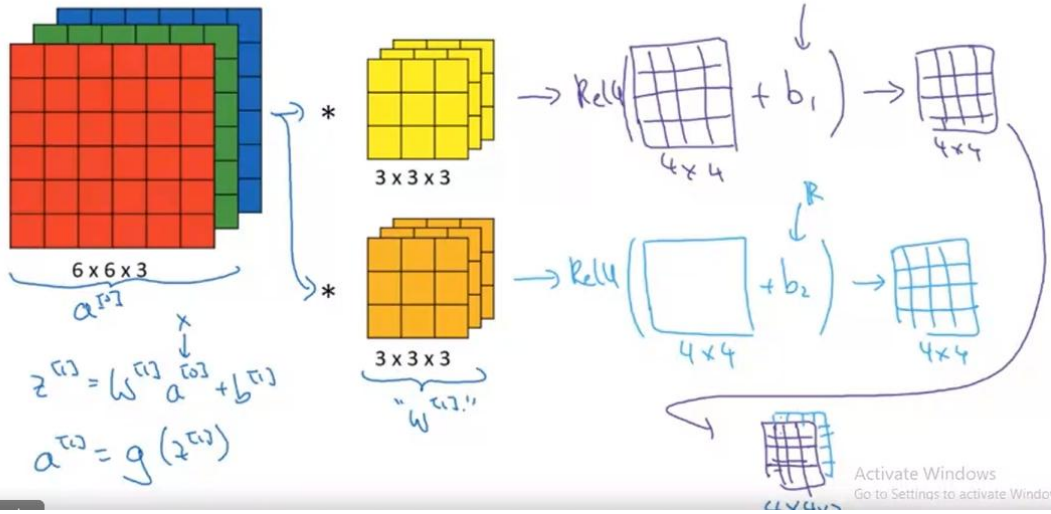
Multiple filters



Summary: $n \times n \times n_c$ * $f \times f \times n_c$



Example of a layer



$z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g(z^{[1]})$