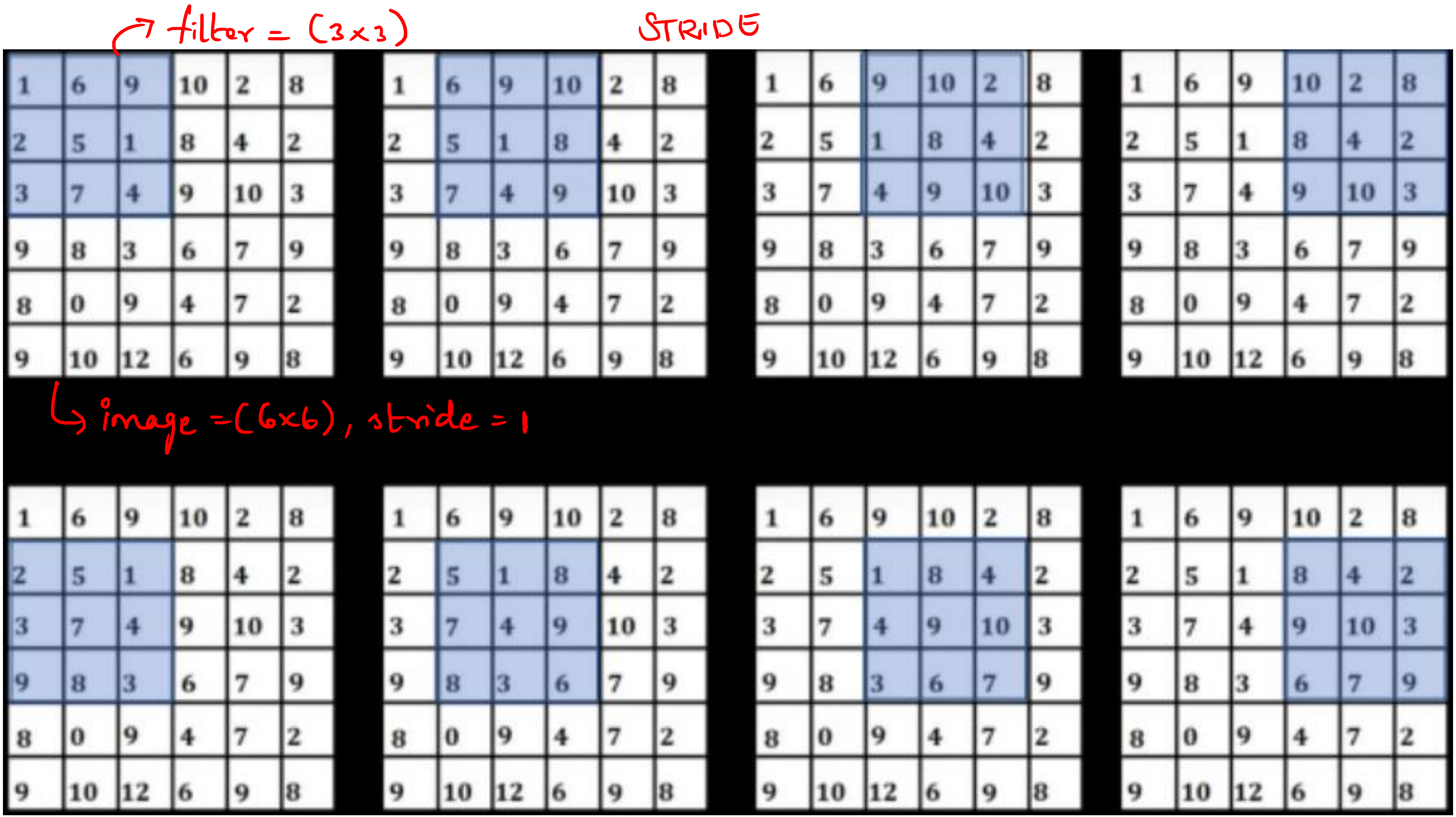
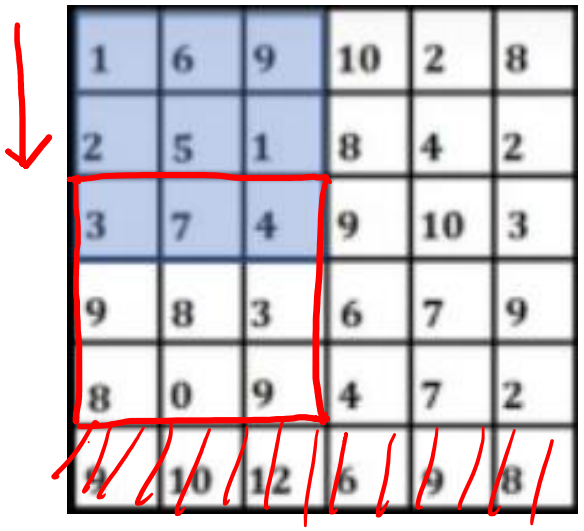
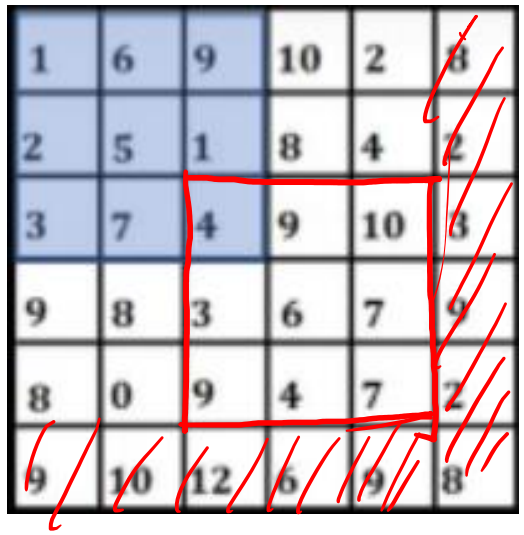

filter = (3x3)
STRIDE
image = (6x6), stride = 1

stride = 2

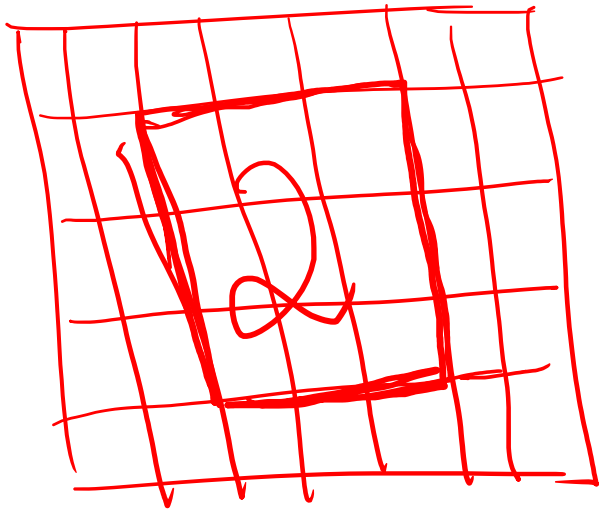
window

1	6	9	10	2	8
2	5	1	8	4	2
3	7	4	9	10	3
9	8	3	6	7	9
8	0	9	4	7	2
9	10	12	6	9	8





This type of convolution is called as strided convolution.



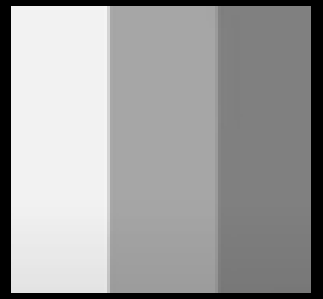
0	0	0	0	0	0	0	0
0	41	5	8	9	2	2	0
0	6	5	1	3	6	5	0
0	1	7	1	2	4	1	0
0	4	2	8	9	2	6	0
0	4	6	1	4	9	7	0
0	1	3	5	6	7	6	0
0	0	0	0	0	0	0	0

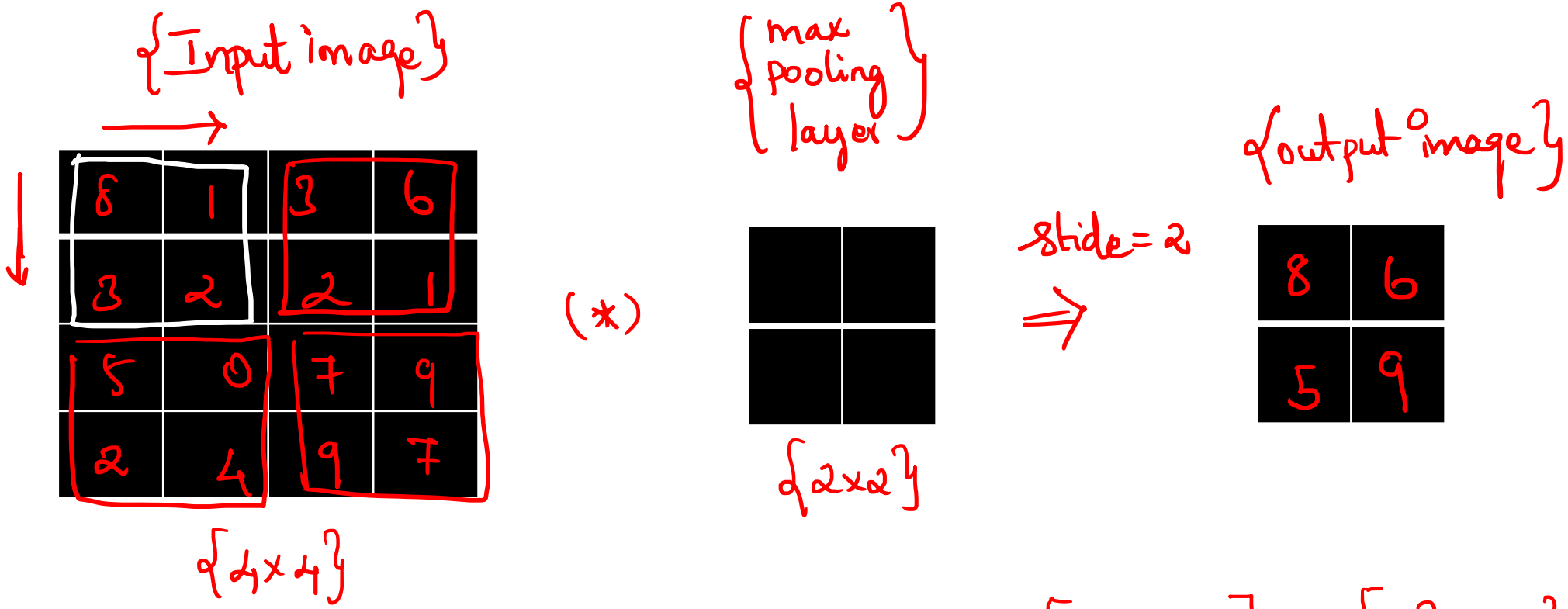
Max-pooling (layer)

Input Image

pooling

Output Image

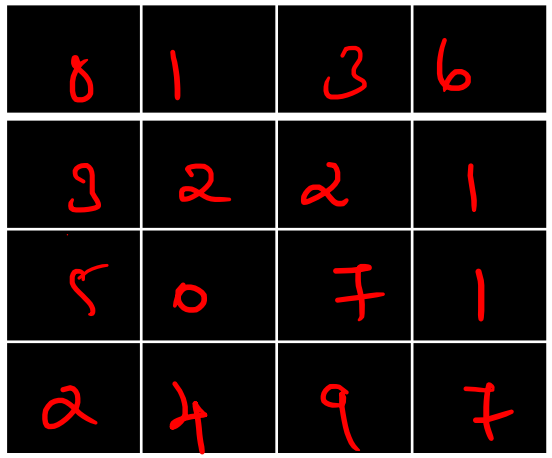




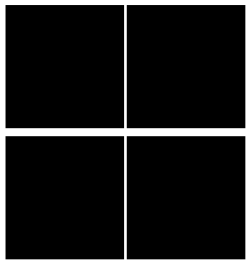
$$(n \times n) \propto (f \times f) = \text{floor}\left[\frac{n-f}{s} + 1\right] \times \text{floor}\left[\frac{n-f}{s} + 1\right] \Rightarrow f\left[\frac{4-2}{2} + 1\right] \Rightarrow f\left[\frac{2}{2} + \frac{2}{2}\right]$$

(s ↓ stride)

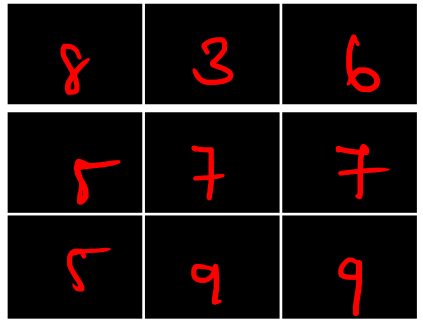
$$\Rightarrow f[4/2] \Rightarrow (2 \times 2)$$



*



stride = 1



[Create duplicate pixel]

$$\text{floor}\left[\frac{n-f}{s}+1\right] \Rightarrow \left[\frac{4-2}{1}+1\right] \Rightarrow 2/1 + 1/1 \rightarrow 3/1 \Rightarrow (3 \times 3)$$

("input Image")

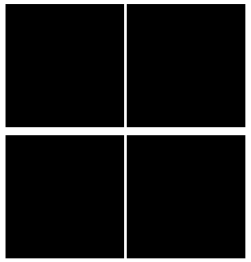
{6×6}

0	0.1	0.2	0.3	0.4	0.5
0	0.1	0.2	0.3	0.4	0.5
0	0.1	0.2	0.3	0.4	0.5
0	0.1	0.2	0.3	0.4	0.5
0	0.1	0.2	0.3	0.4	0.5
0	0.1	0.2	0.3	0.4	0.5

(*)

{maxpooling filter}

{2×2}



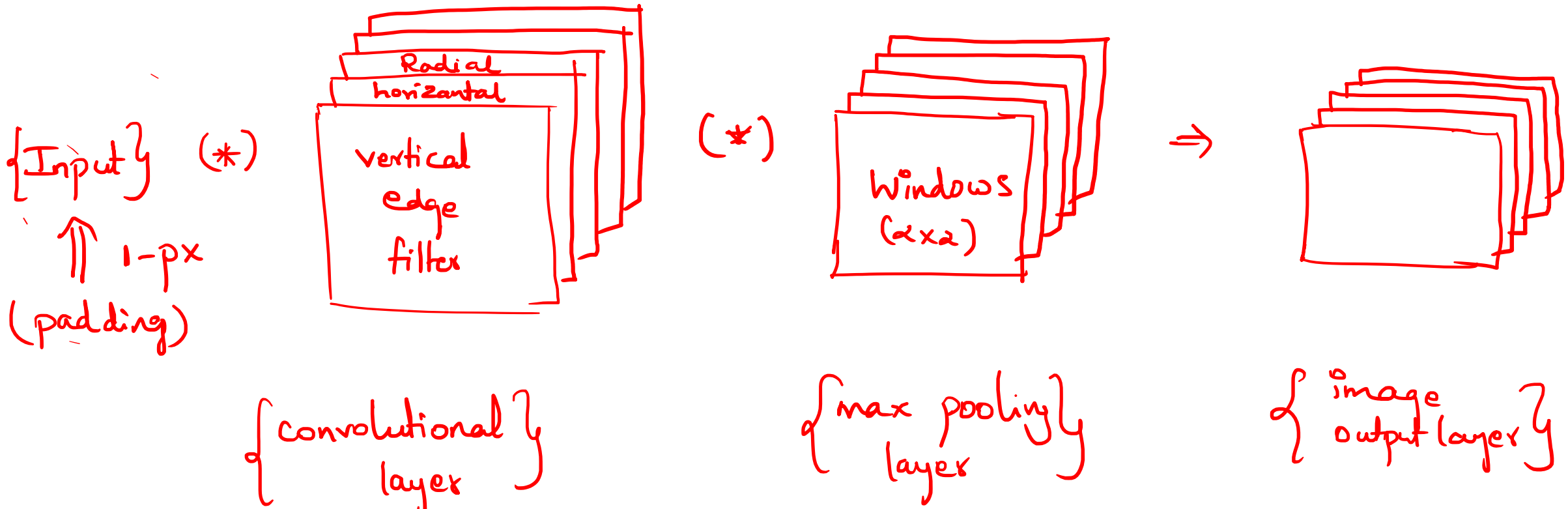
stride = 2

Output image

0.1	0.3	0.5
0.1	0.3	0.5
0.1	0.3	0.5

$$\text{floor}\left[\frac{n-f}{s}+1\right] \Rightarrow f\left[\frac{6-2}{2}+1\right] \Rightarrow f\left[\frac{4}{2}+\frac{2}{2}\right] \Rightarrow (3\times 3)$$

{not mandatory}

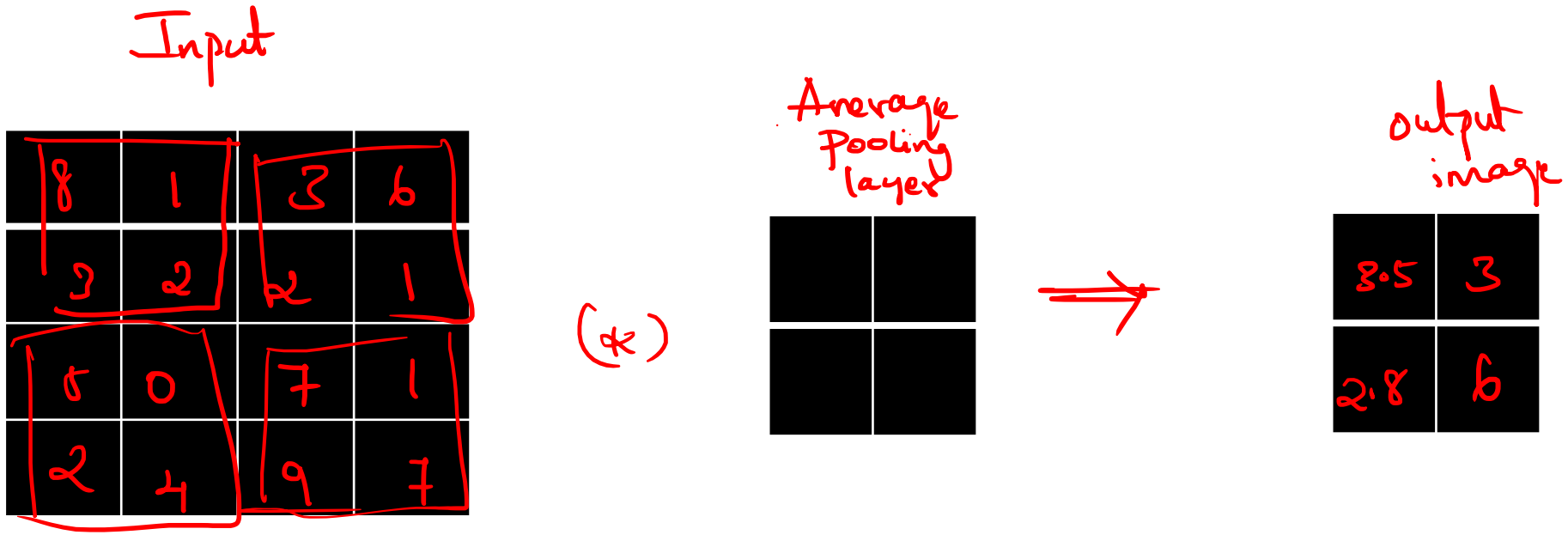


{convolutional layer}

{max pooling layer}

{image output layer}

{Average - pooling}



Why pooling ?

→ Reduce image size to reduce the computational cost

→ Enhance the pixel by maxing it.

Where pooling ?

→ after convolutional layer

→ No parameters involved, no training needed.