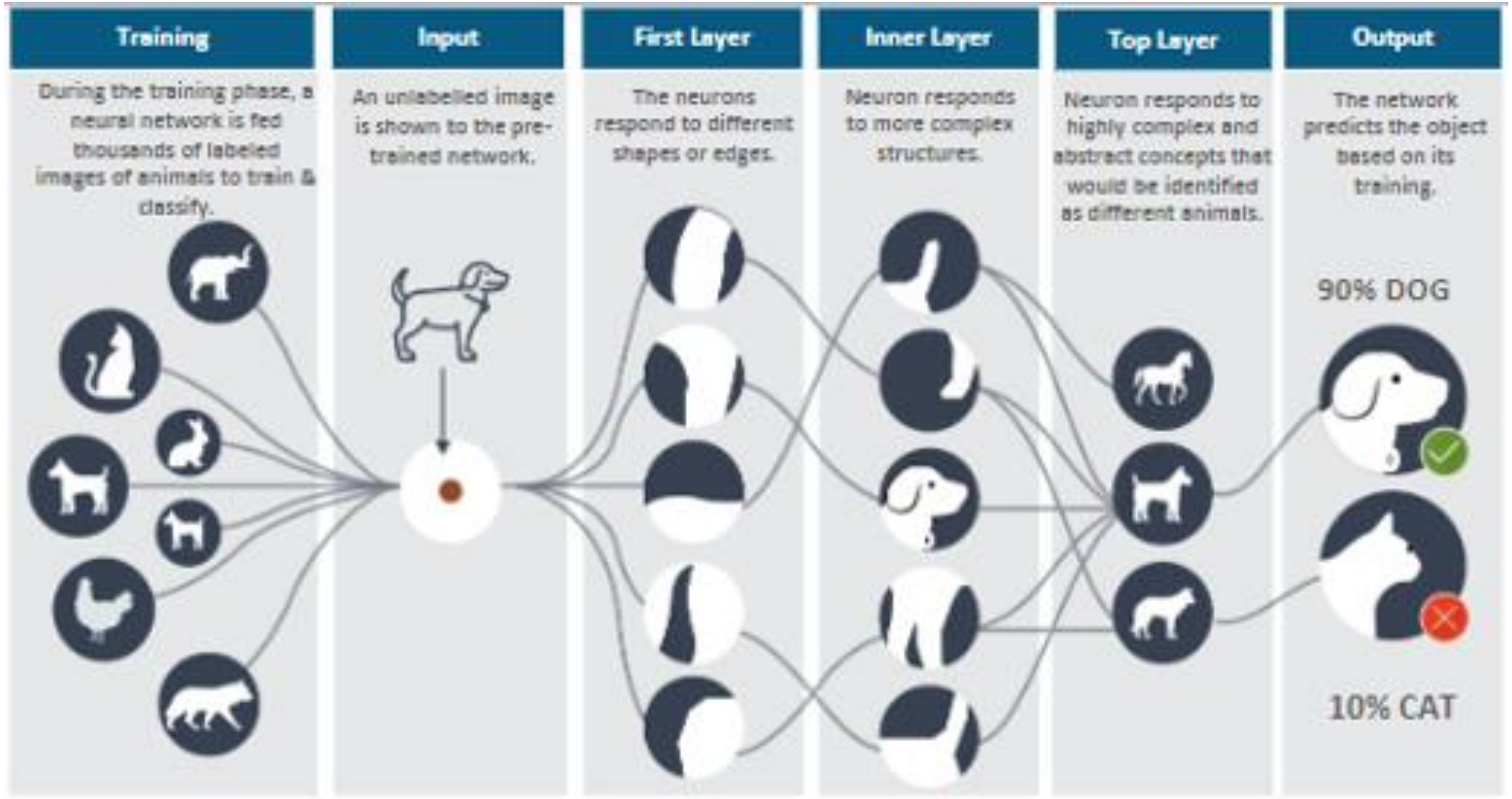
Convolutional Neural Networks



Training

During the training phase, a neural network is fed thousands of labeled images of animals to train & classify.

Input

An unlabelled image is shown to the pre-trained network.

First Layer

The neurons respond to different shapes or edges.

Inner Layer

Neuron responds to more complex structures.

Top Layer

Neuron responds to highly complex and abstract concepts that would be identified as different animals.

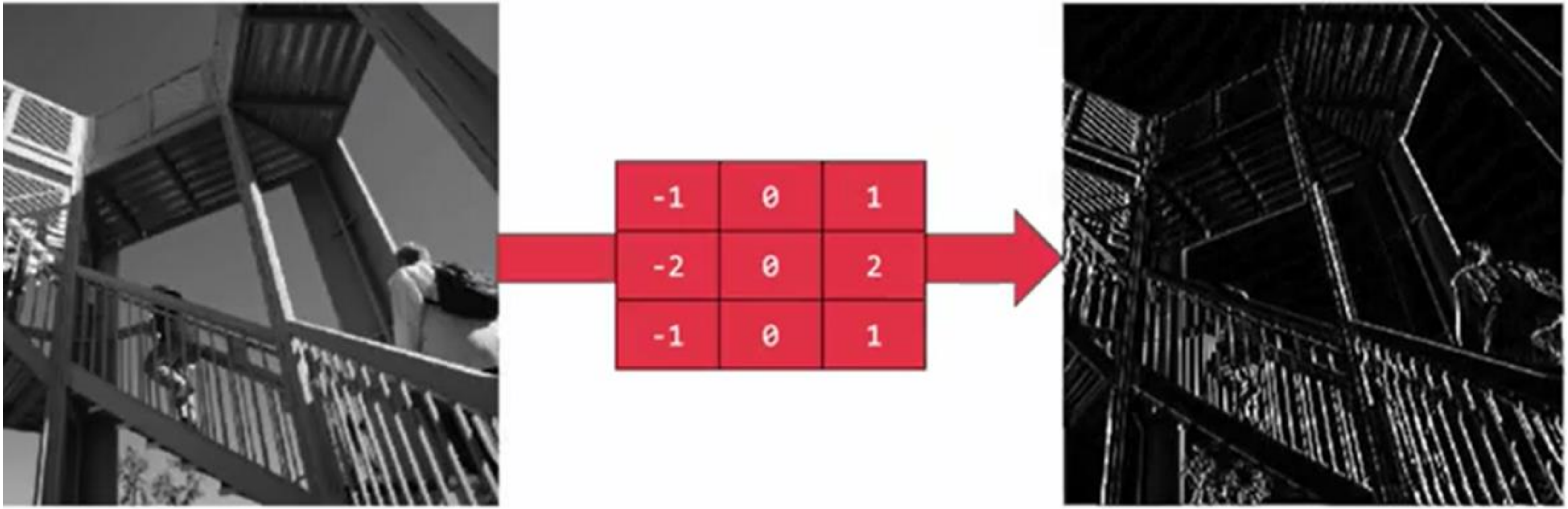
Output

The network predicts the object based on its training.

90% DOG

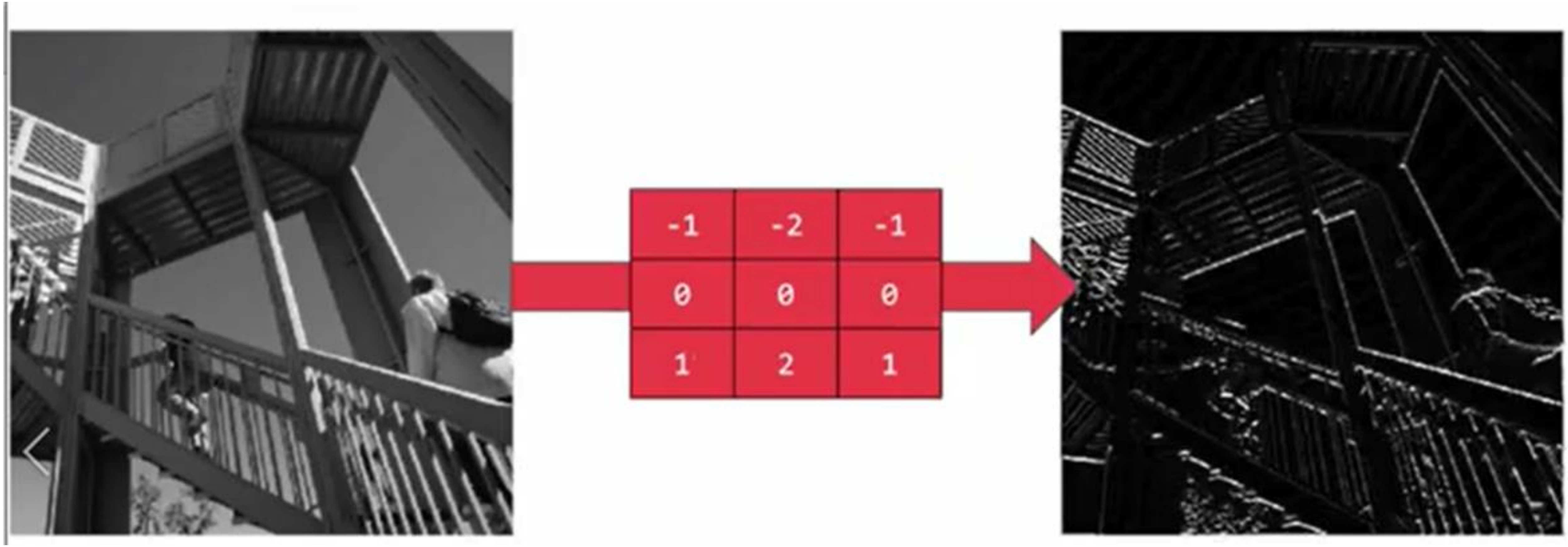
10% CAT

Convolution Operation - Filter bringing out vertical lines



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

Convolution Operation - Filter bringing out horizontal lines

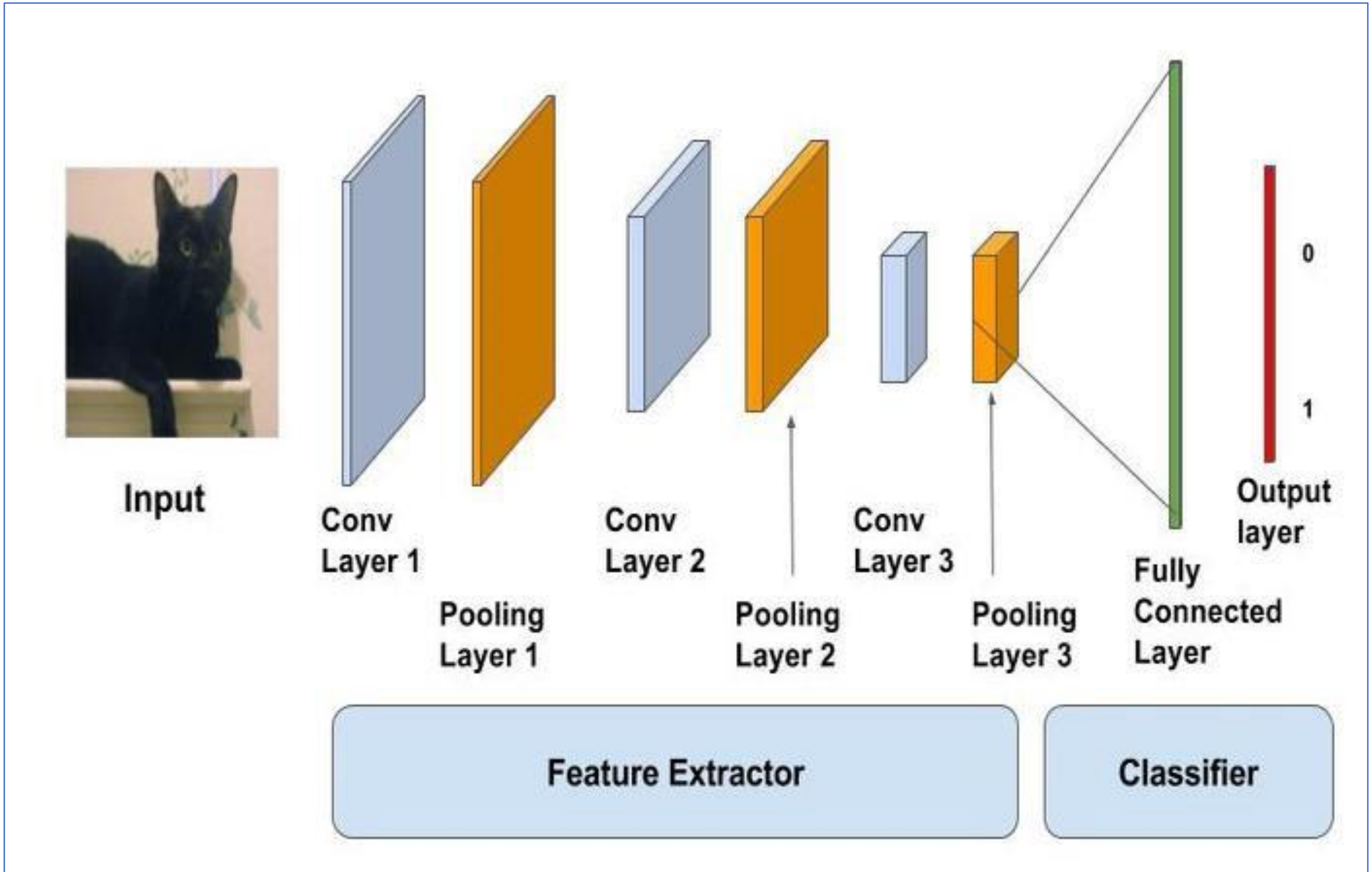


-1	-2	-1
0	0	0
1	2	1

Demo

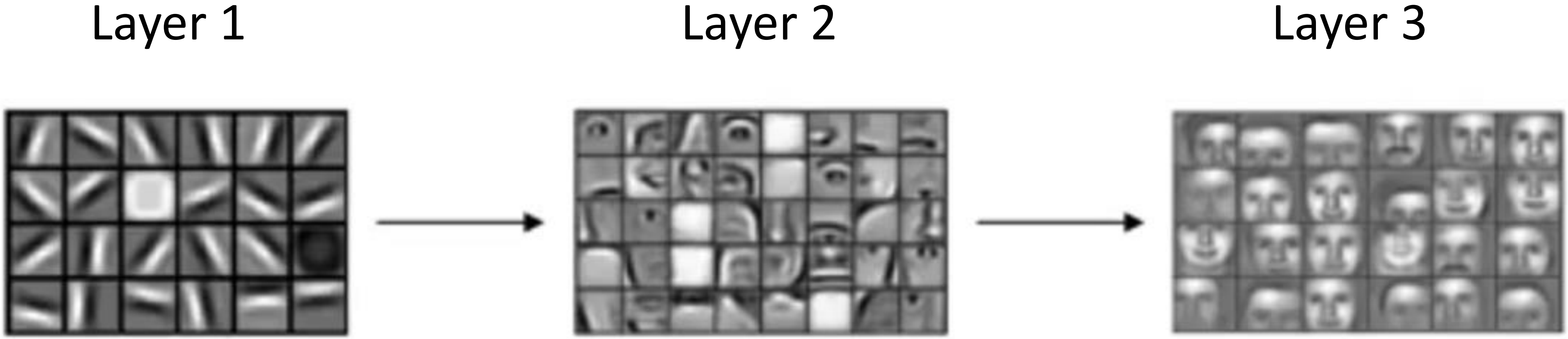
- 00-How convolutions and pooling works.ipynb

Convolutional Neural Network



Convolution operation

- The objective of the Convolution Operation is to **extract the high-level features** such as edges, from the input image.
- ConvNets need not be limited to only one Convolutional Layer.
- Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc.
- With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.

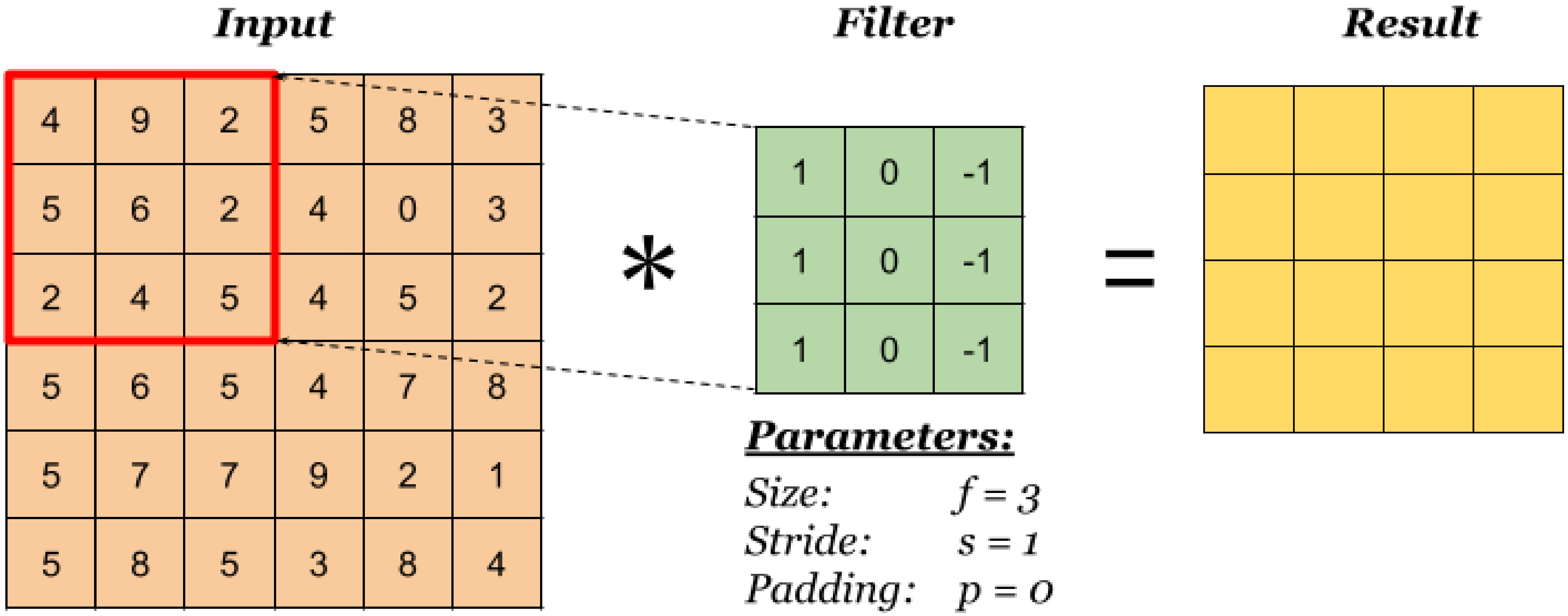


Layer 1

Layer 2

Layer 3

Convolutions



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

$n_H \, x \, n_W = 6 \, x \, 6$

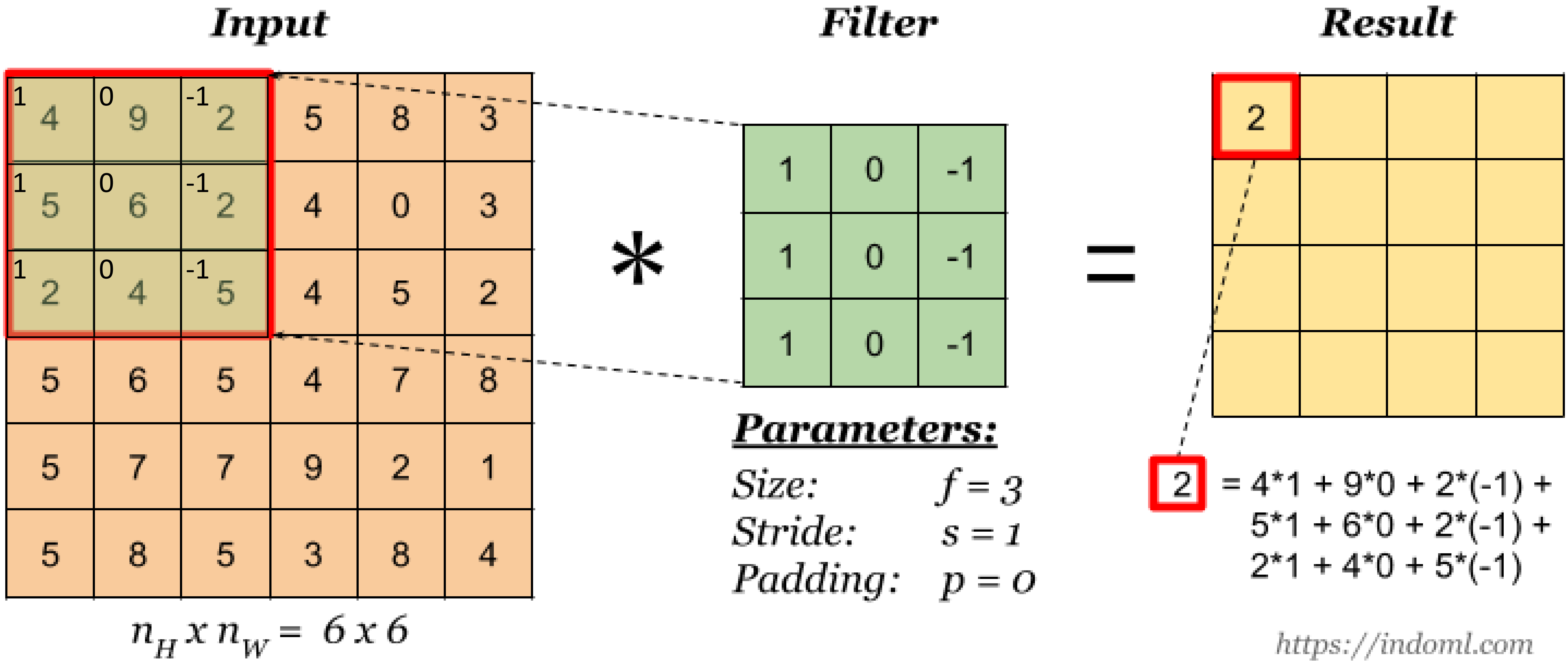
1	0	-1
1	0	-1
1	0	-1

Parameters:

Size: $f = 3$
Stride: $s = 1$
Padding: $p = 0$

https://indoml.com

Convolutions



Input

^{1} 4	^{0} 9	$^{-1}$ 2	5	8	3
^{1} 5	^{0} 6	$^{-1}$ 2	4	0	3
^{1} 2	^{0} 4	$^{-1}$ 5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

$n_H \, x \, n_W = 6 \, x \, 6$

Filter

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

Result

2			

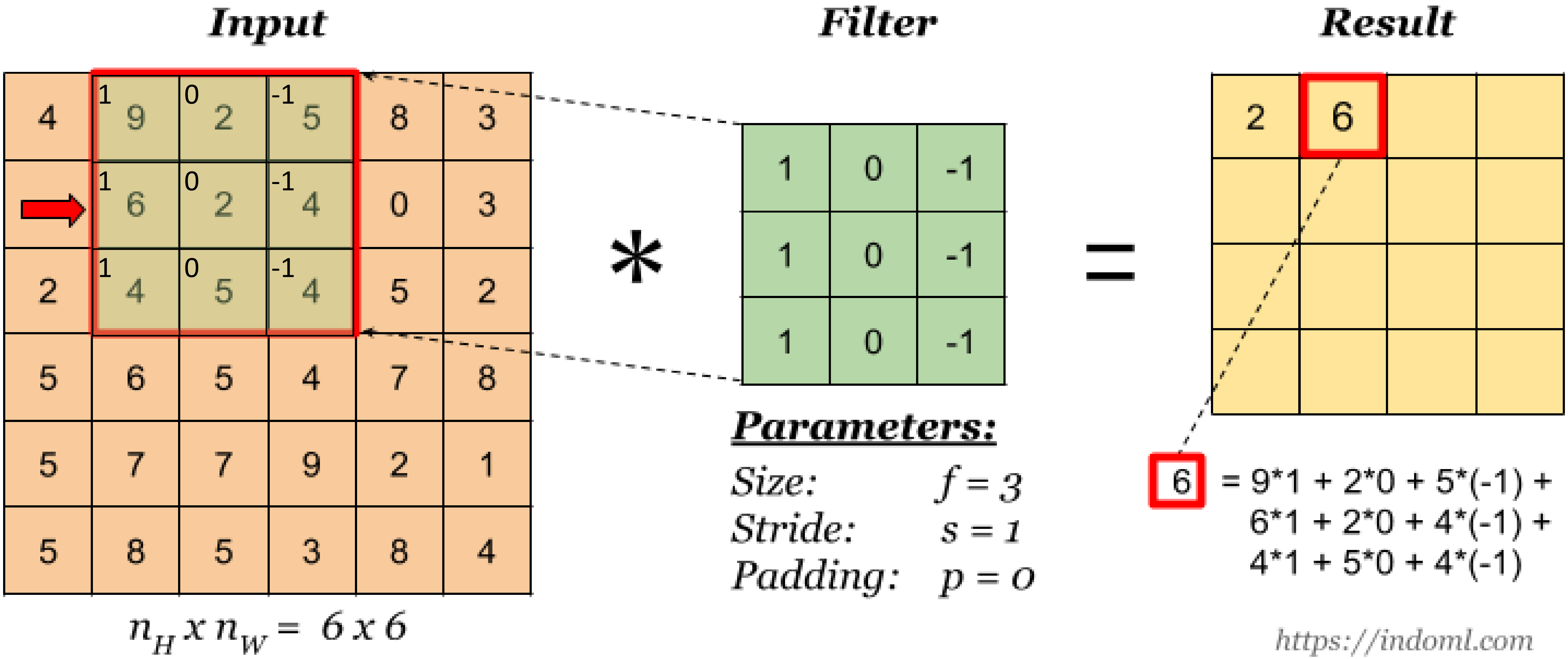
Parameters:

Size: $f = 3$
Stride: $s = 1$
Padding: $p = 0$

2 = 4*1 + 9*0 + 2*(-1) +
5*1 + 6*0 + 2*(-1) +
2*1 + 4*0 + 5*(-1)

https://indoml.com

Convolutions



Input

4	$^{1}9$	$^{0}2$	$^{-1}5$	8	3
2	$^{1}6$	$^{0}2$	$^{-1}4$	0	3
2	$^{1}4$	$^{0}5$	$^{-1}4$	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

$n_H \, x \, n_W = 6 \, x \, 6$

$*$

Filter

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

Parameters:

Size: $f = 3$
Stride: $s = 1$
Padding: $p = 0$

$=$

Result

2	6		

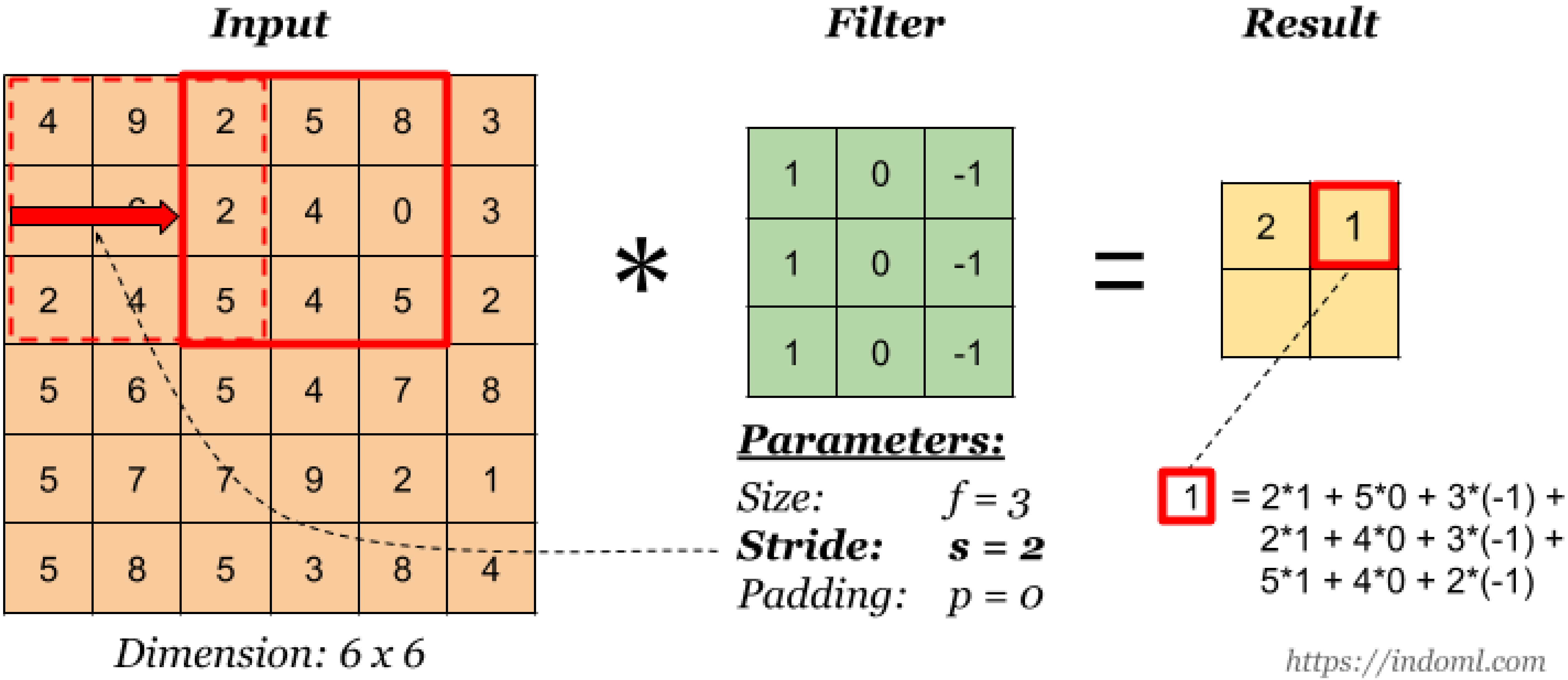
$\boxed{6}$ = 9*1 + 2*0 + 5*(-1) +
6*1 + 2*0 + 4*(-1) +
4*1 + 5*0 + 4*(-1)

https://indoml.com

The total number of multiplications to calculate the result above is (4 x 4) x (3 x 3) = 144.

Convolutions - Strides

Convolutions: Stride = 2



Input

4	9	2	5	8	3
[illegible]	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

Dimension: 6 x 6

*

Filter

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

=

Result

2	1

Parameters:

Size: $f = 3$

Stride: $\boldsymbol{s = 2}$

Padding: $p = 0$

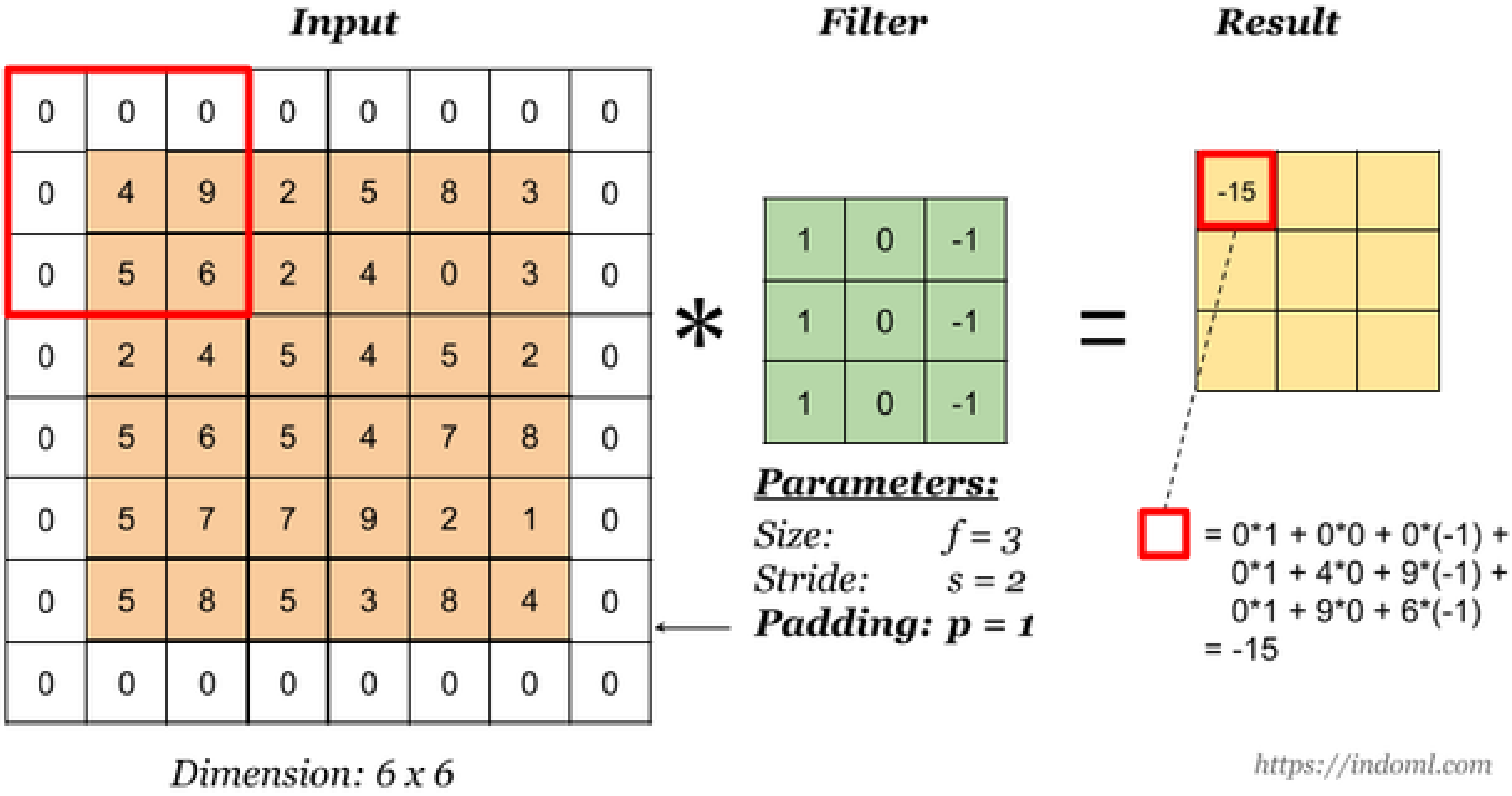
1 = 2*1 + 5*0 + 3*(-1) +
2*1 + 4*0 + 3*(-1) +
5*1 + 4*0 + 2*(-1)

https://indoml.com

The total number of multiplications to calculate the result above is (2 x 2) x (3 x 3) = 36

Convolutions - Padding

Convolutions: Padding = 1



Input

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

Dimension: 6 x 6

*

Filter

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

=

Result

-15		

Parameters:

Size: $f = 3$
Stride: $s = 2$
Padding: $p = 1$

= 0*1 + 0*0 + 0*(-1) +
0*1 + 4*0 + 9*(-1) +
0*1 + 9*0 + 6*(-1)
= -15

https://indoml.com

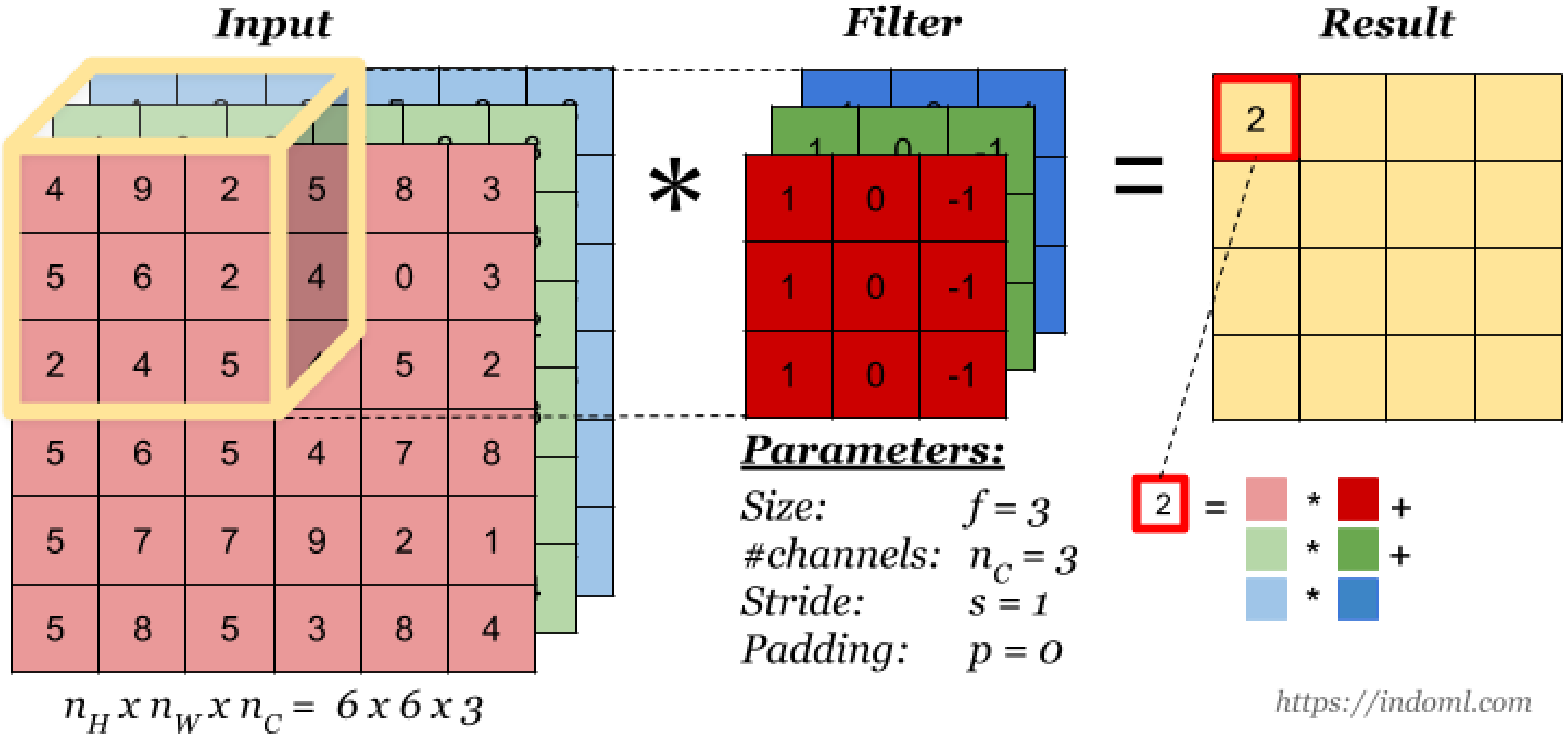
Resultant dimensions

- (n – f + 2p)/s + 1
  - n is the original dimension
  - f is the filter dimension
  - p is the padding
  - s is the strides
- For a 6x6 image, if we have a filter 3x3, padding =0, stride=1
  - Resultant dimensions will be: (6 – 3 + 2*0)/1 + 1 = 4
  - 4x4
- For a 6x6 image, if we have a filter 3x3, padding =1, stride=1
  - Resultant dimensions will be: (6 – 3 + 2*1)/1 + 1 = 6
  - 6x6

- “**valid**” padding : no padding
- “**same**” padding: output dimension does not change

Convolutions – Multiple Input Channels

Convolutions: Image with RGB channels (ch > 1)



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

$n_H \, x \, n_W \, x \, n_C = 6 \, x \, 6 \, x \, 3$

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

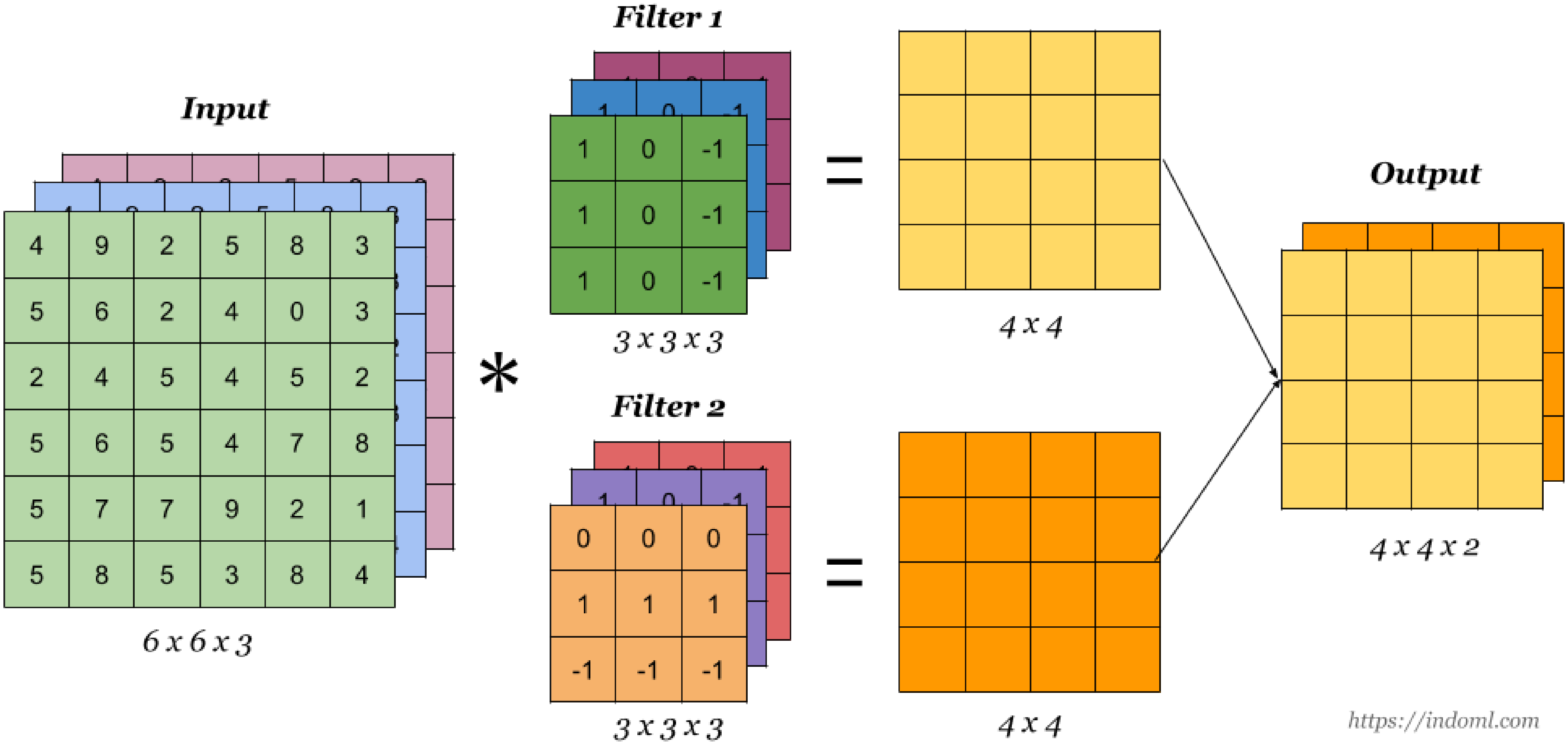
Parameters:

Size: $f = 3$
#channels: $n_C = 3$
Stride: $s = 1$
Padding: $p = 0$

The total number of multiplications to calculate the result is (4 x 4) x (3 x 3 x 3) = 432

Convolutions – Multiple Filters

Convolutions: Multiple channels, multiple filters



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

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

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

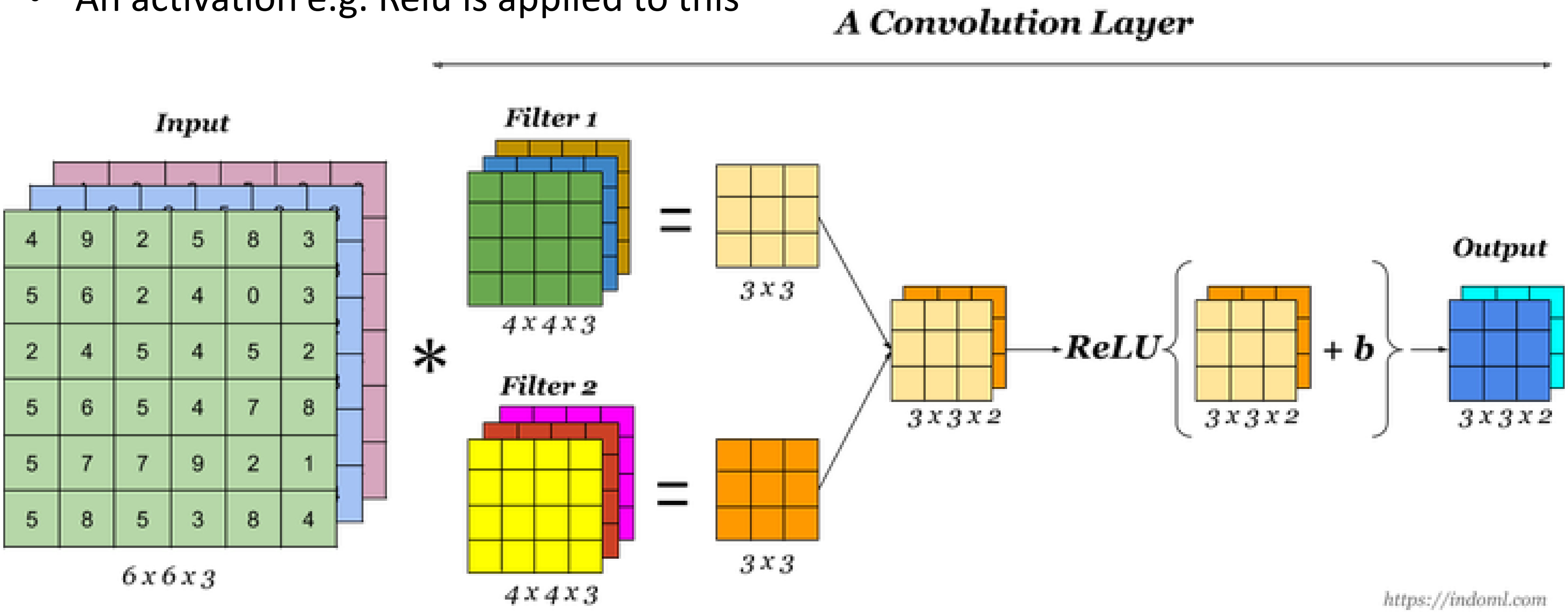
The total number of multiplications to calculate the result is (4 x 4 x 2) x (3 x 3 x 3) = 864

A Convolution Layer

A Convolution Layer

A convolution layer is made up of:

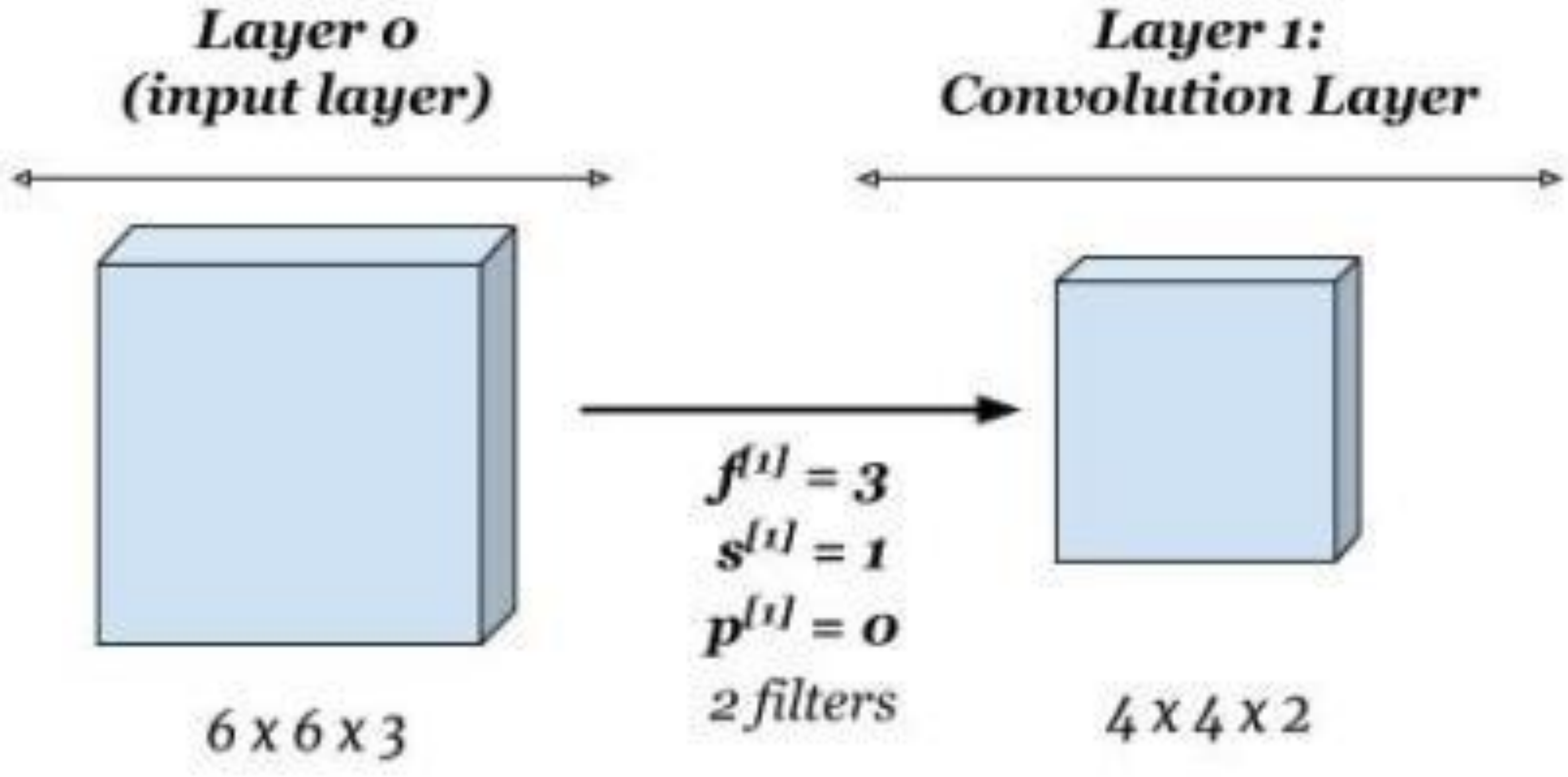
- The convolution we saw earlier
- A bias is then added to this convolution
- An activation e.g. Relu is applied to this



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

https://indoml.com

Convolution Layer – simpler representation



Layer 0 (input layer)

Layer 1: Convolution Layer

$f^{[1]} = 3$

$s^{[1]} = 1$

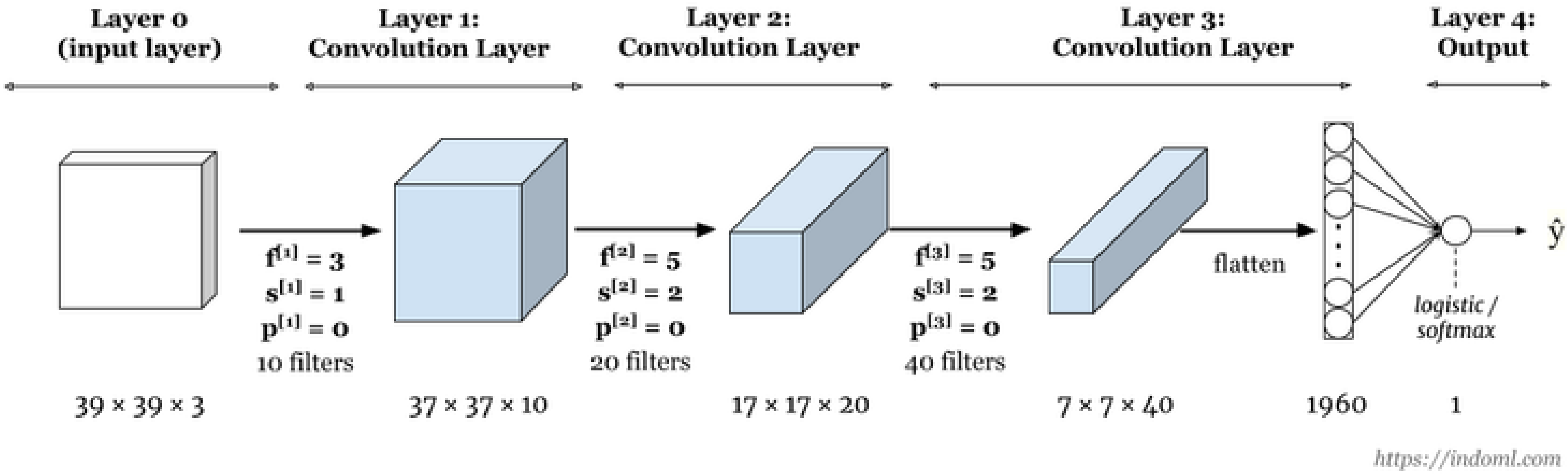
$p^{[1]} = 0$

2 filters

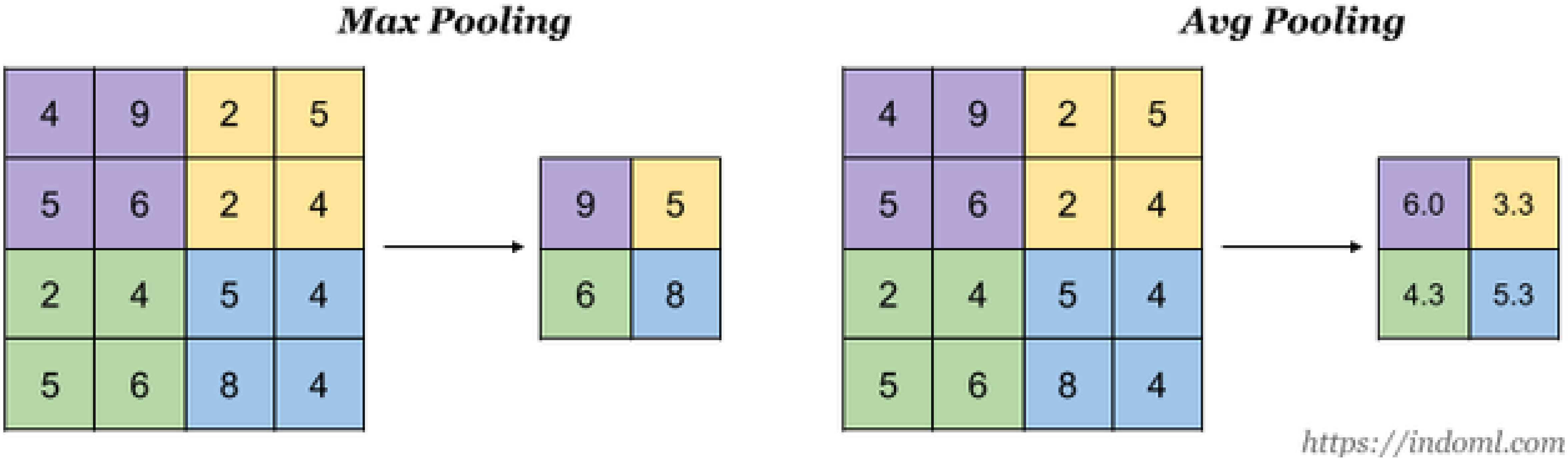
6 x 6 x 3

4 x 4 x 2

Multiple Convolution Layers



Pooling



Max Pooling

4	9	2	5
5	6	2	4
2	4	5	4
5	6	8	4

9	5
6	8

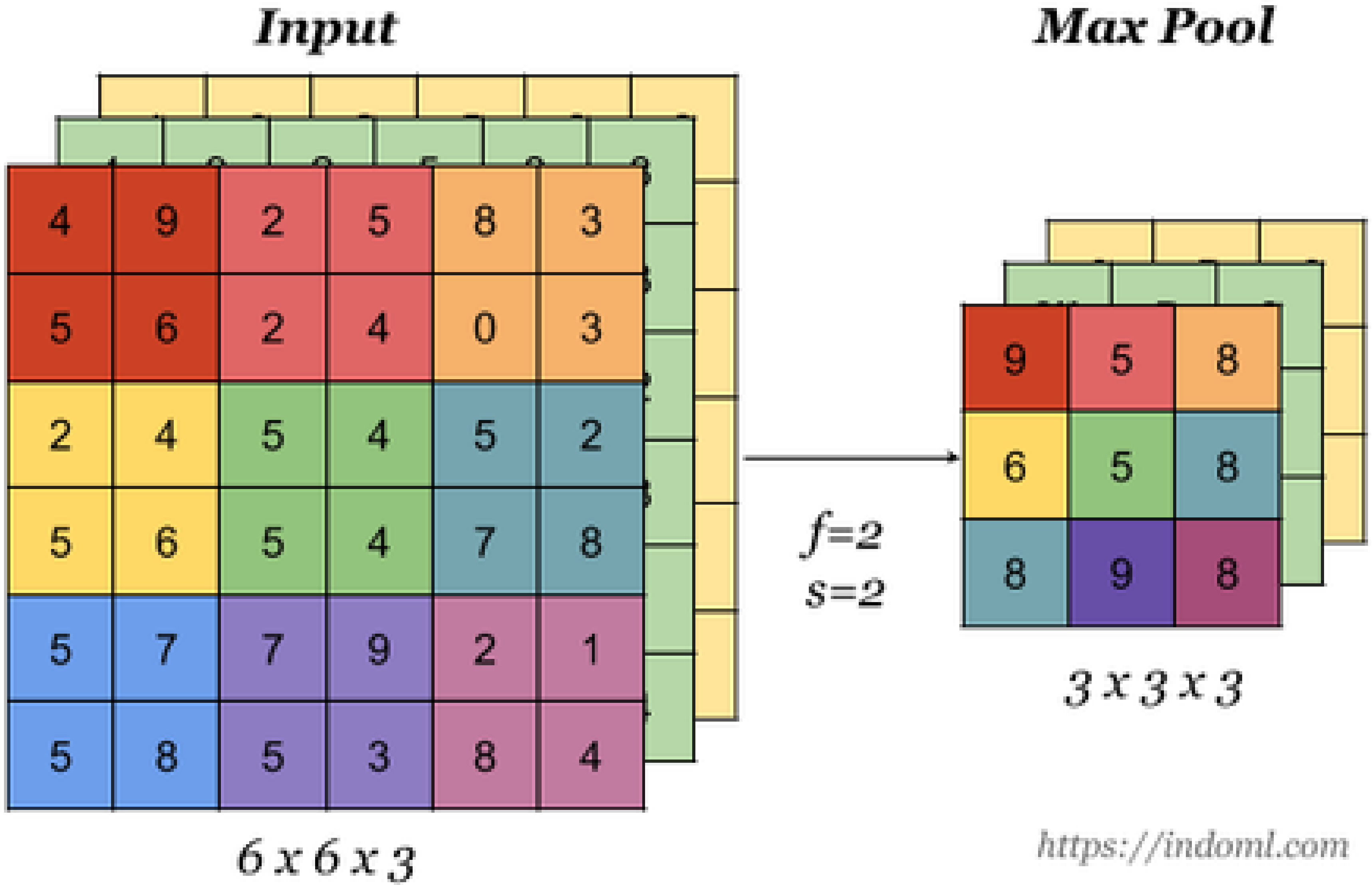
Avg Pooling

4	9	2	5
5	6	2	4
2	4	5	4
5	6	8	4

6.0	3.3
4.3	5.3

https://indoml.com

Max Pooling



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

9	5	8
6	5	8
8	9	8

https://indoml.com

Tensorflow Conv2D layer - params to learn

e.g.

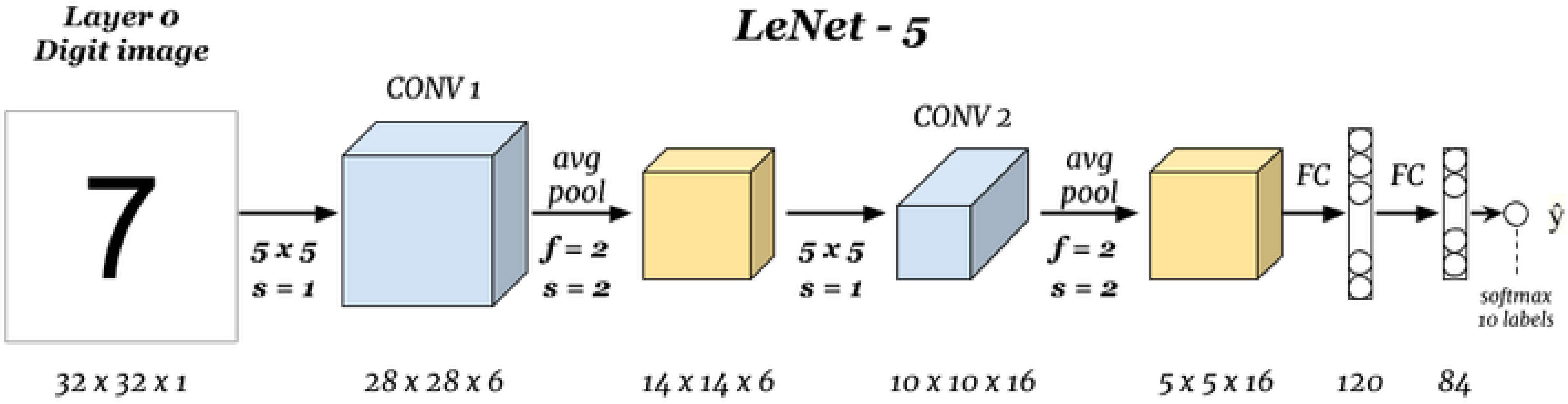
model = Sequential()

model.add(Conv2D(64,(3,3), input_shape=input_shape)) #64 filters with 3*3 filter

- Input_shape -> shape of the image input to the Conv2D layer
- If input_shape = (150,150,3), each of the 64 filters will be of size (3,3,3)
  - Note that each filter will have the same number of channels as the input image
- Thus, this Conv2D layer will need to learn 64 filters each of size (3,3,3), which means 1792 params

A Convolutional Network

Le-Net 5 Network


LeNet - 5
Layer 0
Digit image
7
32 x 32 x 1
5 x 5
s = 1
CONV 1
28 x 28 x 6
avg
pool
f = 2
s = 2
14 x 14 x 6
5 x 5
s = 1
CONV 2
10 x 10 x 16
avg
pool
f = 2
s = 2
5 x 5 x 16
FC
120
FC
84
ŷ
softmax
10 labels