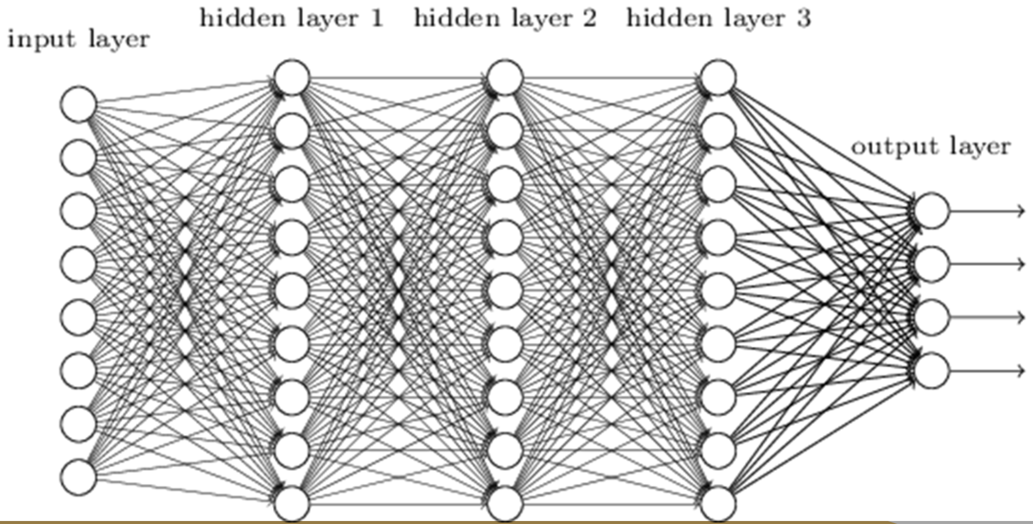
# Convolution Neural Network

Chapter-3

## Why CNN?

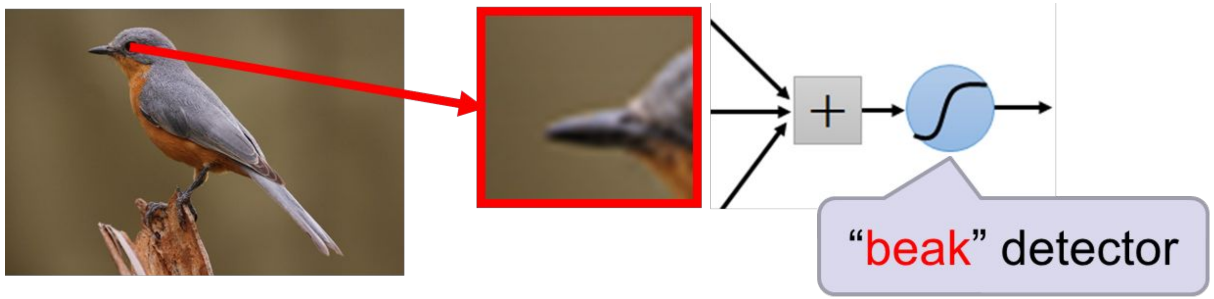
- We know it is good to learn a small model.
- From this fully connected model, do we really need all the edges?
- Can some of these be shared?



## Why CNN?

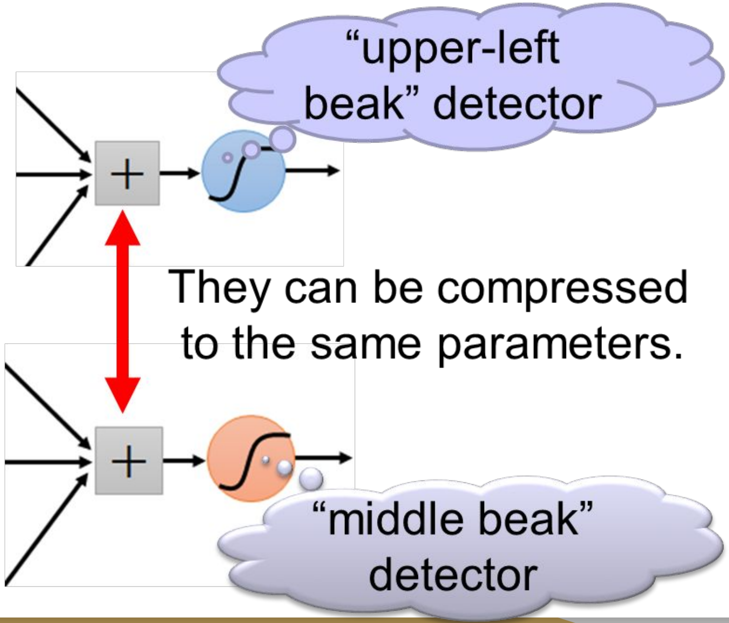
- Some patterns are much smaller than the whole image

  Can represent a small region with fewer parameters



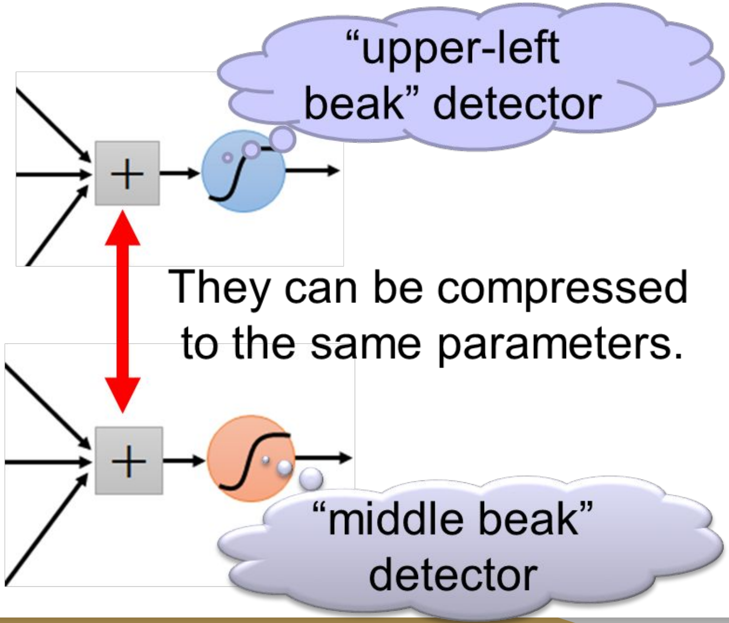
# Why CNN?

- Same pattern appears in different places:
- They can be compressed!
- What about training a lot of such “small” detectors and each detector must “move around”.


“upper-left beak” detector
They can be compressed to the same parameters.
“middle beak” detector

## Why CNN?

- Same pattern appears in different places:
- They can be compressed!
- What about training a lot of such “small” detectors and each detector must “move around”.

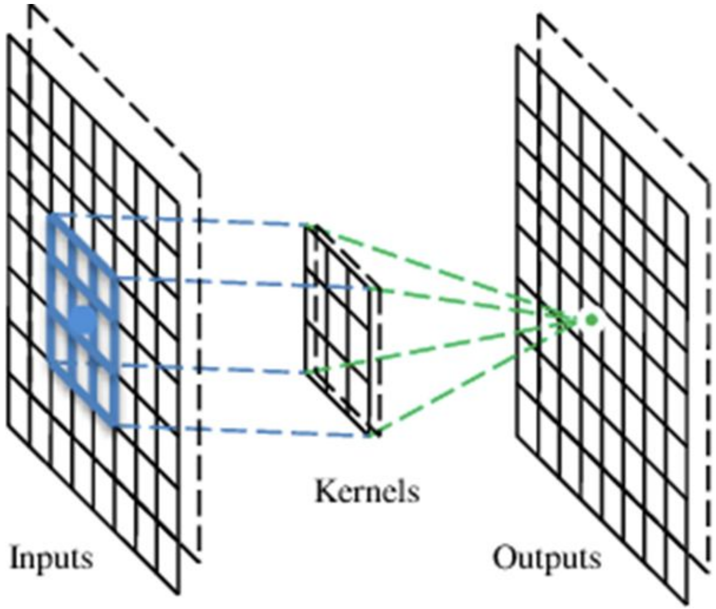

“upper-left beak” detector
They can be compressed to the same parameters.
“middle beak” detector

## Convolutional Networks

- specialized kind of neural network for processing data that has a known, grid-like topology.
- Examples include time-series data, which can be thought of as a 1D grid taking samples at regular time intervals, and image data, which can be thought of as a 2D grid of pixels.
- The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution.
- ***“Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.”***

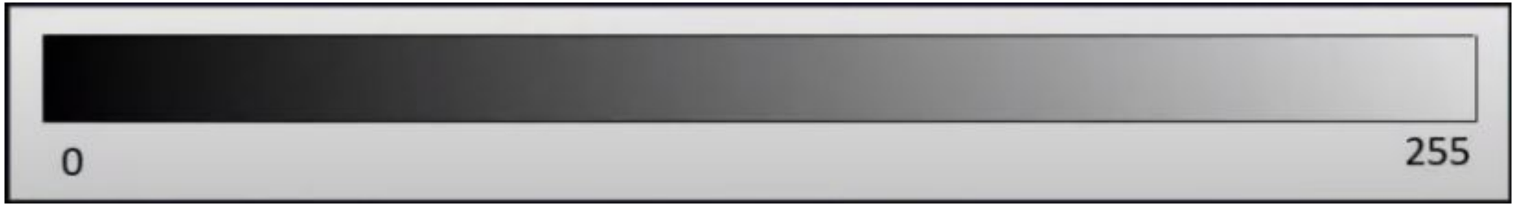
# A convolutional layer

- A CNN is a neural network with some convolutional layers (and some other layers).
- A convolutional layer has a number of filters that does convolutional operation.

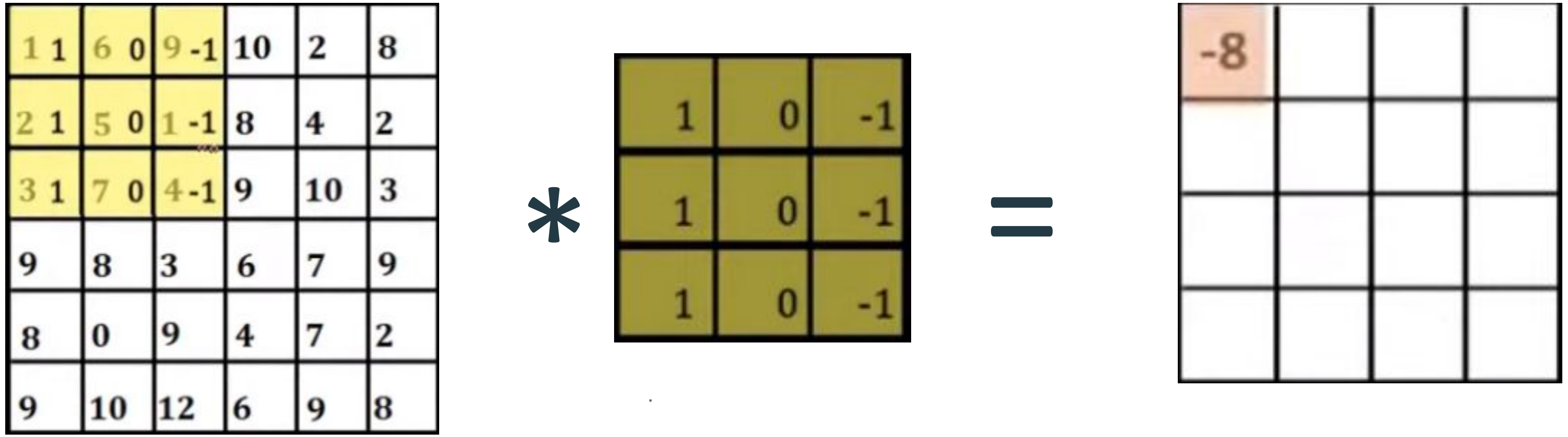


# The Convolution Operation

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 6 & 9 & 10 & 2 & 8 \\ \hline 2 & 5 & 1 & 8 & 4 & 2 \\ \hline 3 & 7 & 4 & 9 & 10 & 3 \\ \hline 9 & 8 & 3 & 6 & 7 & 9 \\ \hline 8 & 0 & 9 & 4 & 7 & 2 \\ \hline 9 & 10 & 12 & 6 & 9 & 8 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline \phantom{0} & \phantom{0} & \phantom{0} & \phantom{0} \\ \hline \phantom{0} & & & \\ \hline \phantom{0} & & & \\ \hline \phantom{0} & & & \\ \hline \end{array}$$



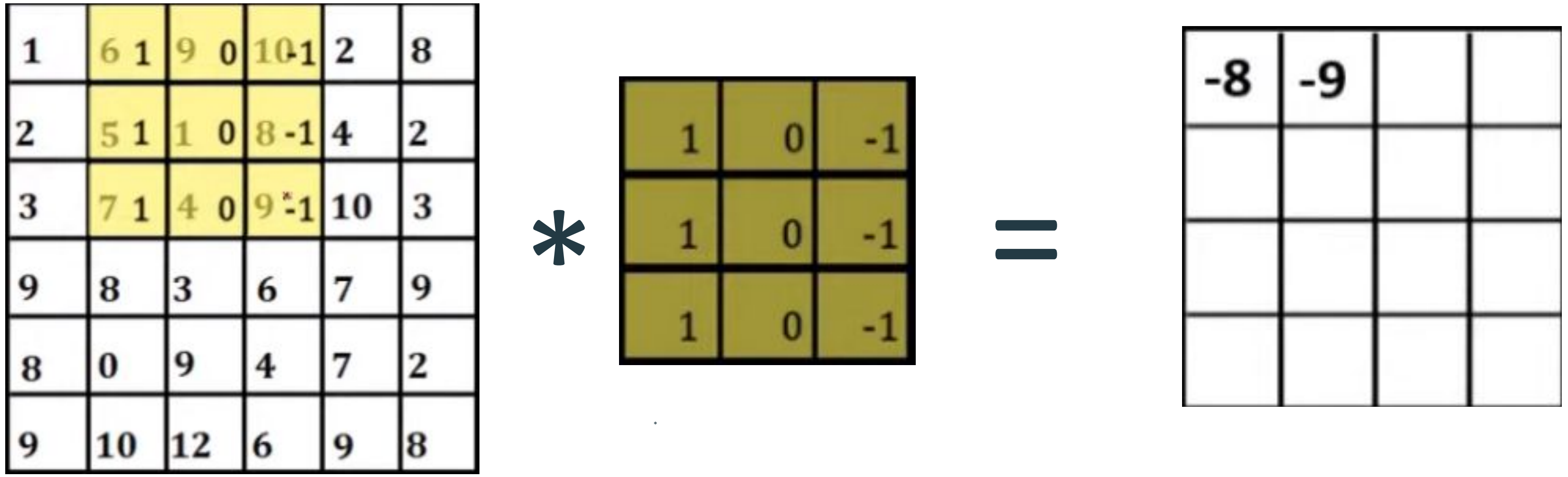
# The Convolution Operation



$$\begin{array}{|c|c|c|c|c|c|}\hline 1\ 1 & 6\ 0 & 9\ {-1} & 10 & 2 & 8 \\ \hline 2\ 1 & 5\ 0 & 1\ {-1} & 8 & 4 & 2 \\ \hline 3\ 1 & 7\ 0 & 4\ {-1} & 9 & 10 & 3 \\ \hline 9 & 8 & 3 & 6 & 7 & 9 \\ \hline 8 & 0 & 9 & 4 & 7 & 2 \\ \hline 9 & 10 & 12 & 6 & 9 & 8 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline -8 & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

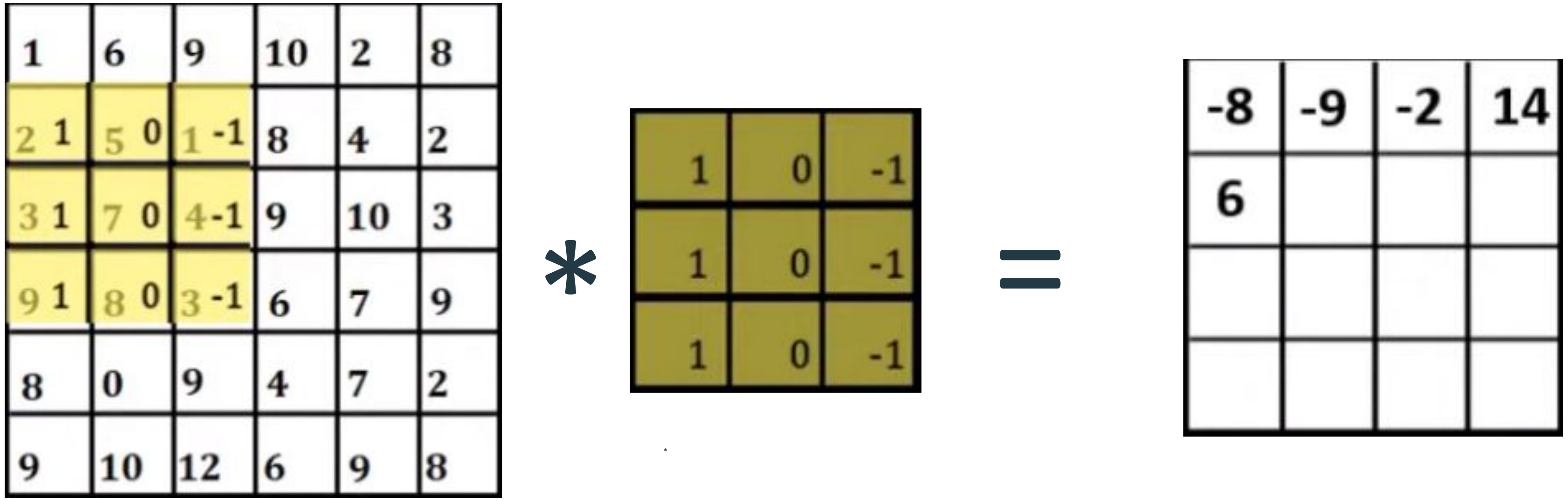
$$1*1 + 2*1 + 3*1 + 6*0 + 5*0 + 7*0 + 9*(-1) + 1*(-1) + 4*(-1) = -8$$

# The Convolution Operation



6*1 + 5*1 + 7*1 + 9*0 + 1*0 + 4*0 + 10*(-1) + 8*(-1) + 9 *(-1) = -9

# The Convolution Operation



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

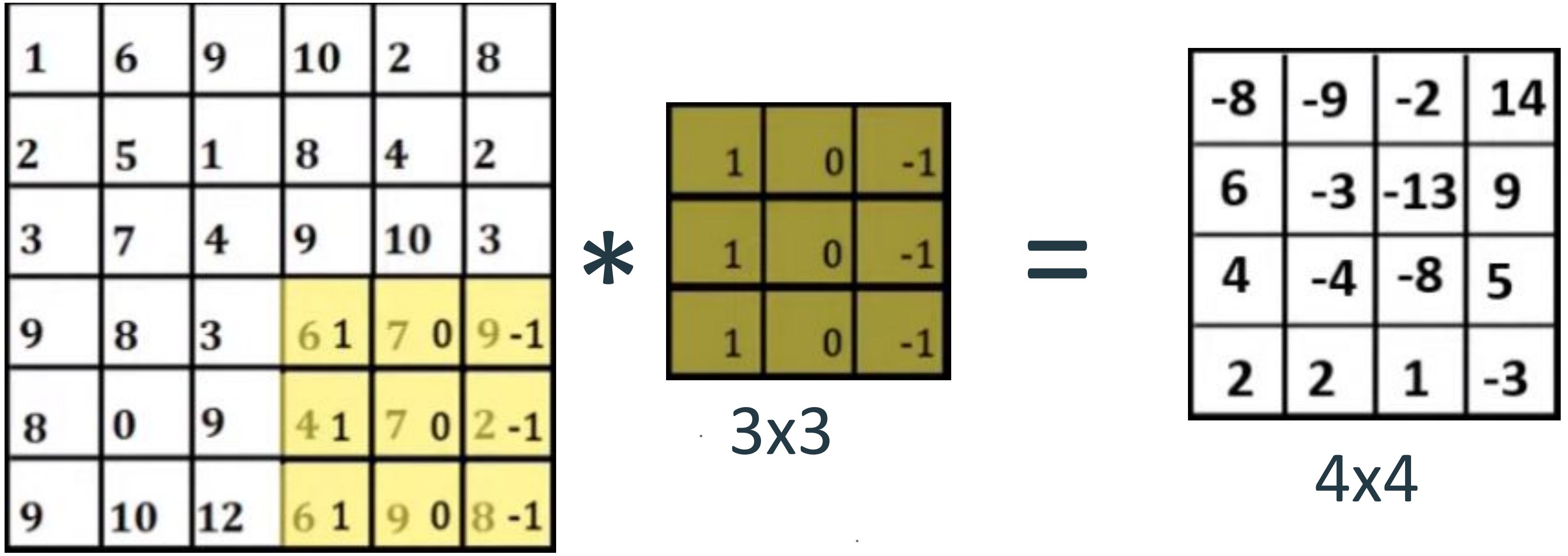
$*$

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

$=$

| | | | |
|---|---|---|---|
| -8 | -9 | -2 | 14 |
| 6 | | | |
| | | | |
| | | | |

## The Convolution Operation



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

6x6

*

|  |  |  |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

3x3

=

|  |  |  |  |
|---|---|---|---|
| -8 | -9 | -2 | 14 |
| 6 | -3 | -13 | 9 |
| 4 | -4 | -8 | 5 |
| 2 | 2 | 1 | -3 |

4x4

$$(n x n)*(f x f)=(n-f+1)x(n-f+1)$$

# The Convolution Operation

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

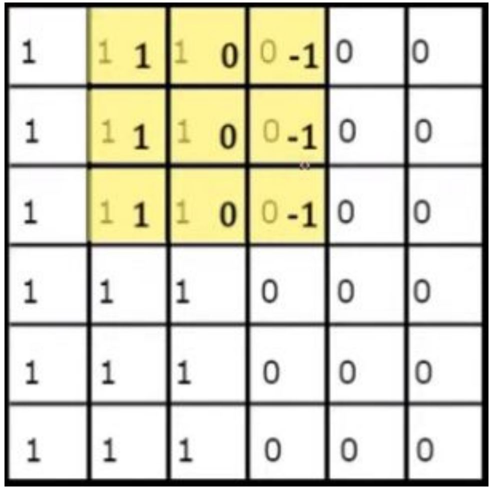
$*$

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

$=$

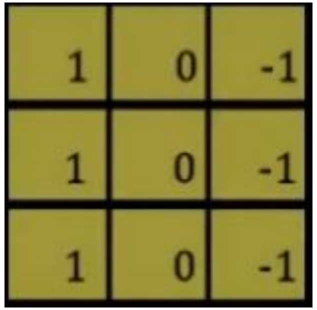
1*1 + 1*1 + 1*1
+ 1*0 + 1*0 + 1*0
+ 1*(-1) + 1*(-1) + 1*(-1) = 0

# The Convolution Operation



| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 1 | 1 0 | 0 -1 | 0 | 0 |
| 1 | 1 1 | 1 0 | 0 -1 | 0 | 0 |
| 1 | 1 1 | 1 0 | 0 -1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

$*$



| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

$=$

$$1*1 + 1*1 + 1*1 + 1*0 + 1*0 + 1*0 + 0*(-1) + 0*(-1) + 0*(-1) = 3$$

## The Convolution Operation

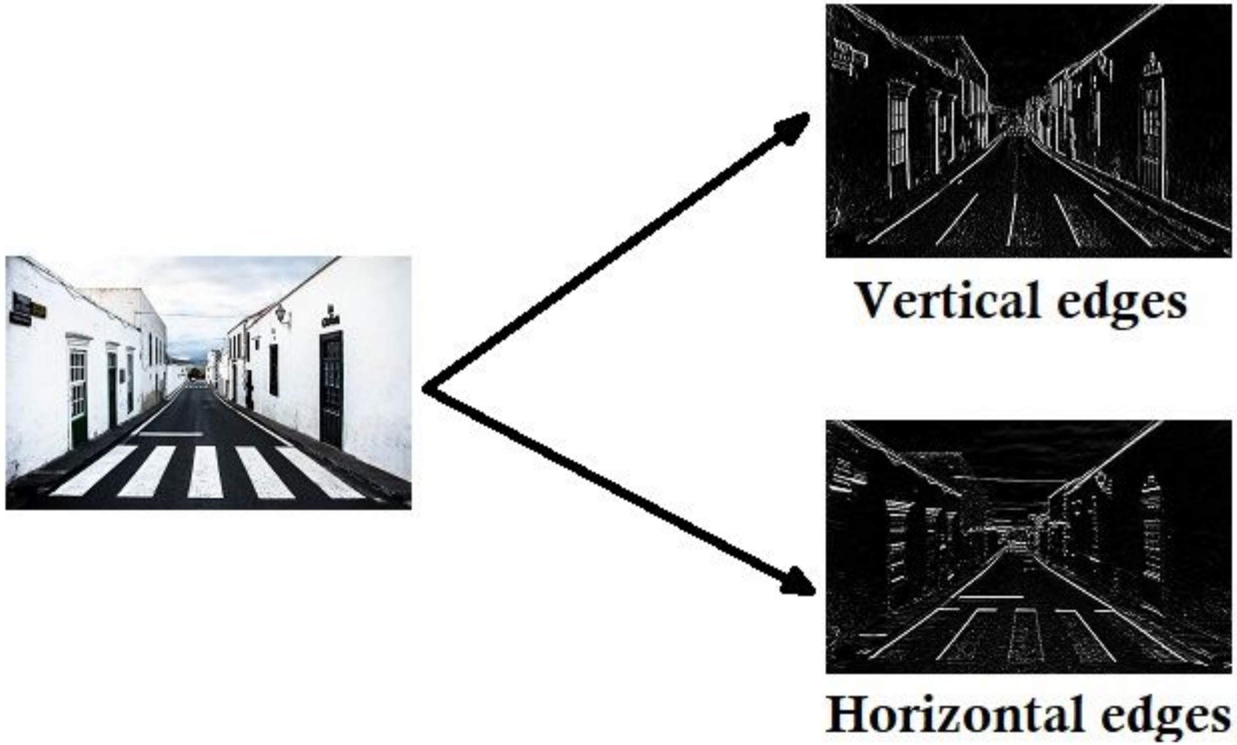
| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 1 | 1 0 | 0 -1 | 0 | 0 |
| 1 | 1 1 | 1 0 | 0 -1 | 0 | 0 |
| 1 | 1 1 | 1 0 | 0 -1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

$*$

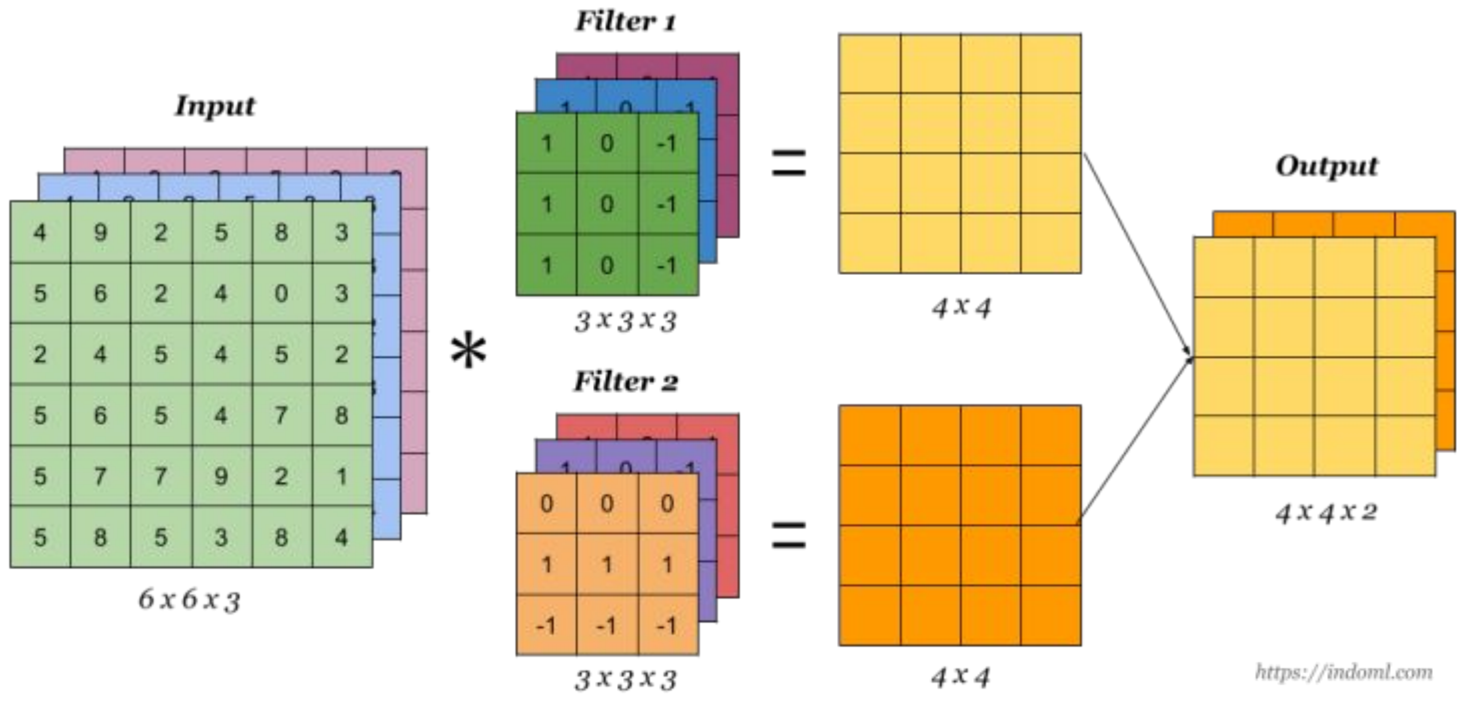
| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

$=$

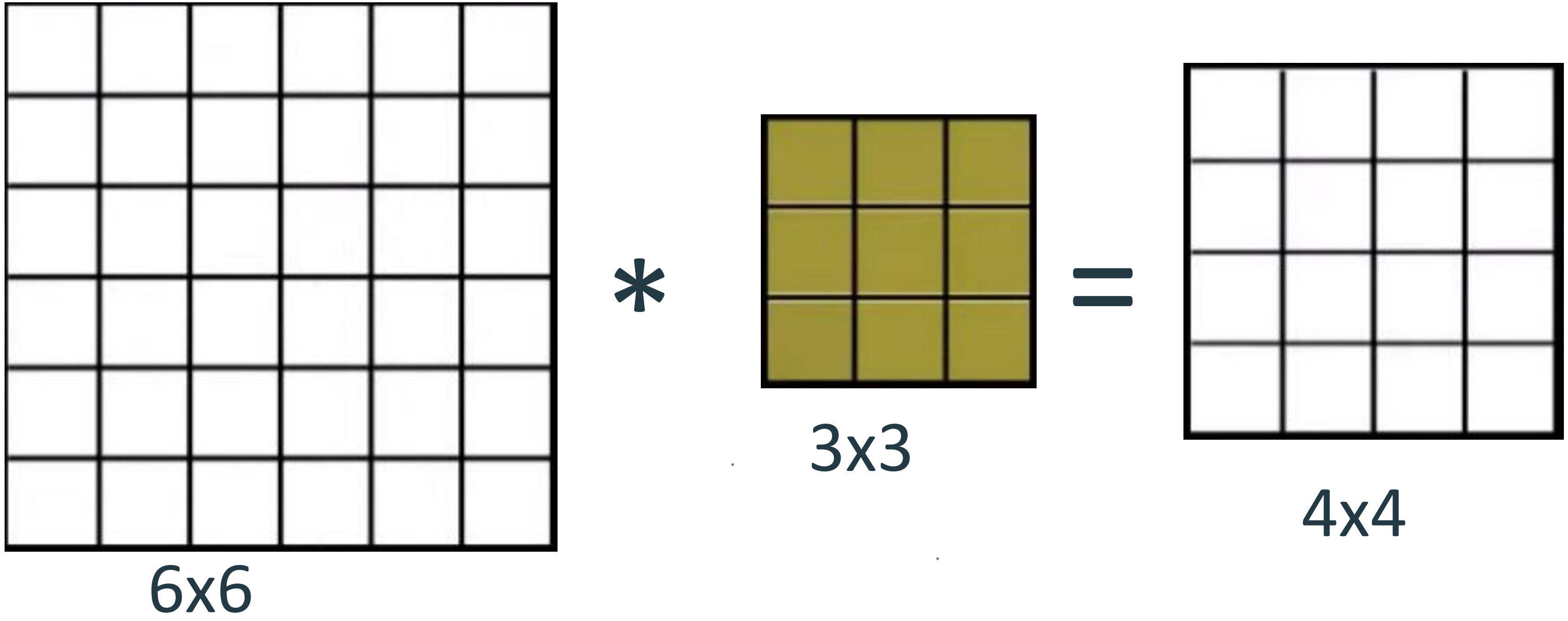
| | | | |
|---|---|---|---|
| 0 | 3 | 3 | 0 |
| 0 | 3 | 3 | 0 |
| 0 | 3 | 3 | 0 |
| 0 | 3 | 3 | 0 |


Vertical edges
Horizontal edges

# Convolution on RBG scale



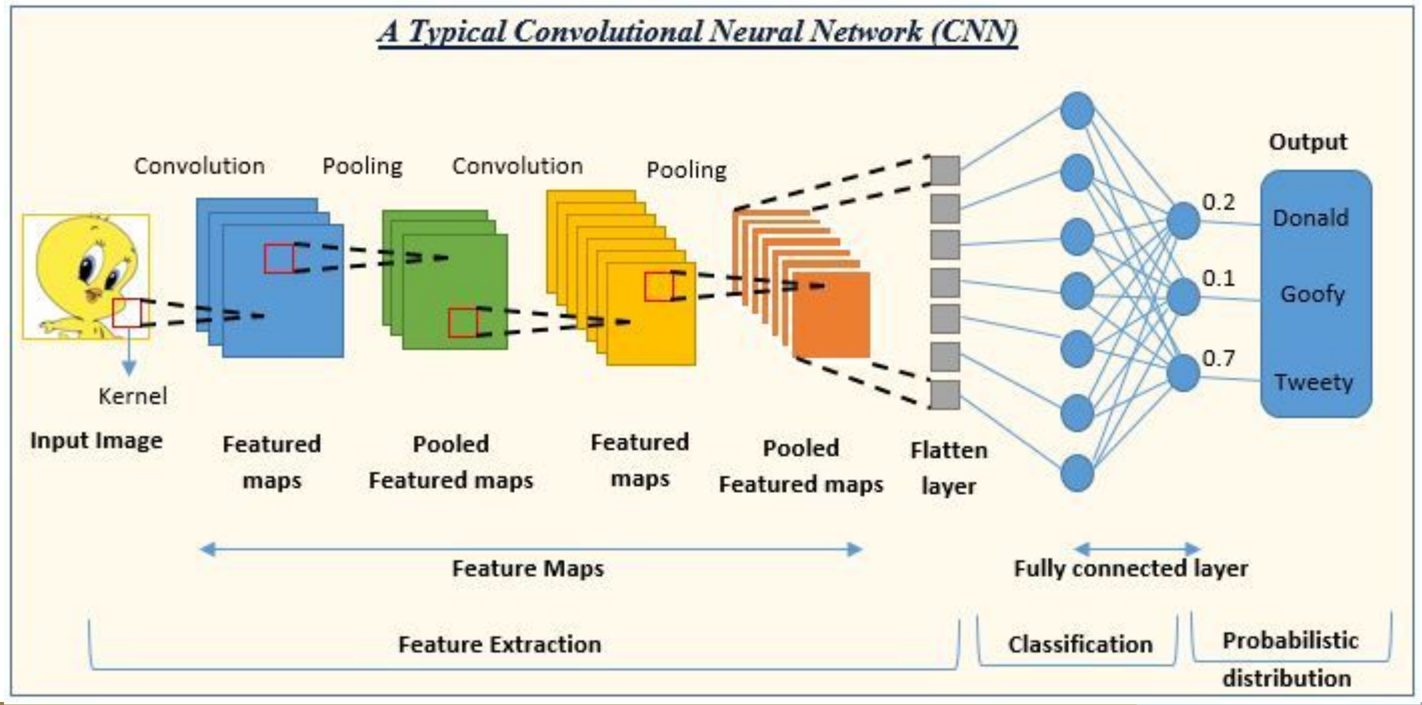
## Padding



## Padding

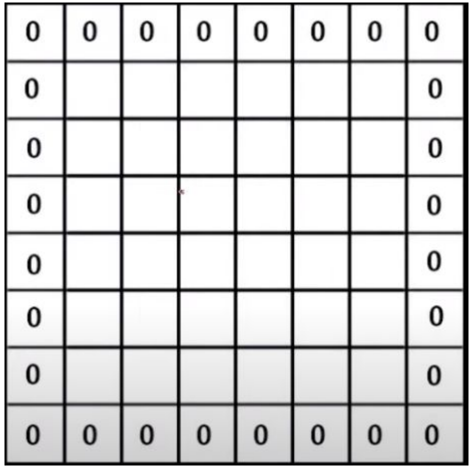
- One observation is that the convolution operation reduces the size of the convolved layer in comparison with the size of the input layer.
- This type of reduction in size is not desirable in general, because it tends to lose some information along the borders of the image (or of the feature map, in the case of hidden layers).
- This problem can be resolved by using padding.

# Typical CNN

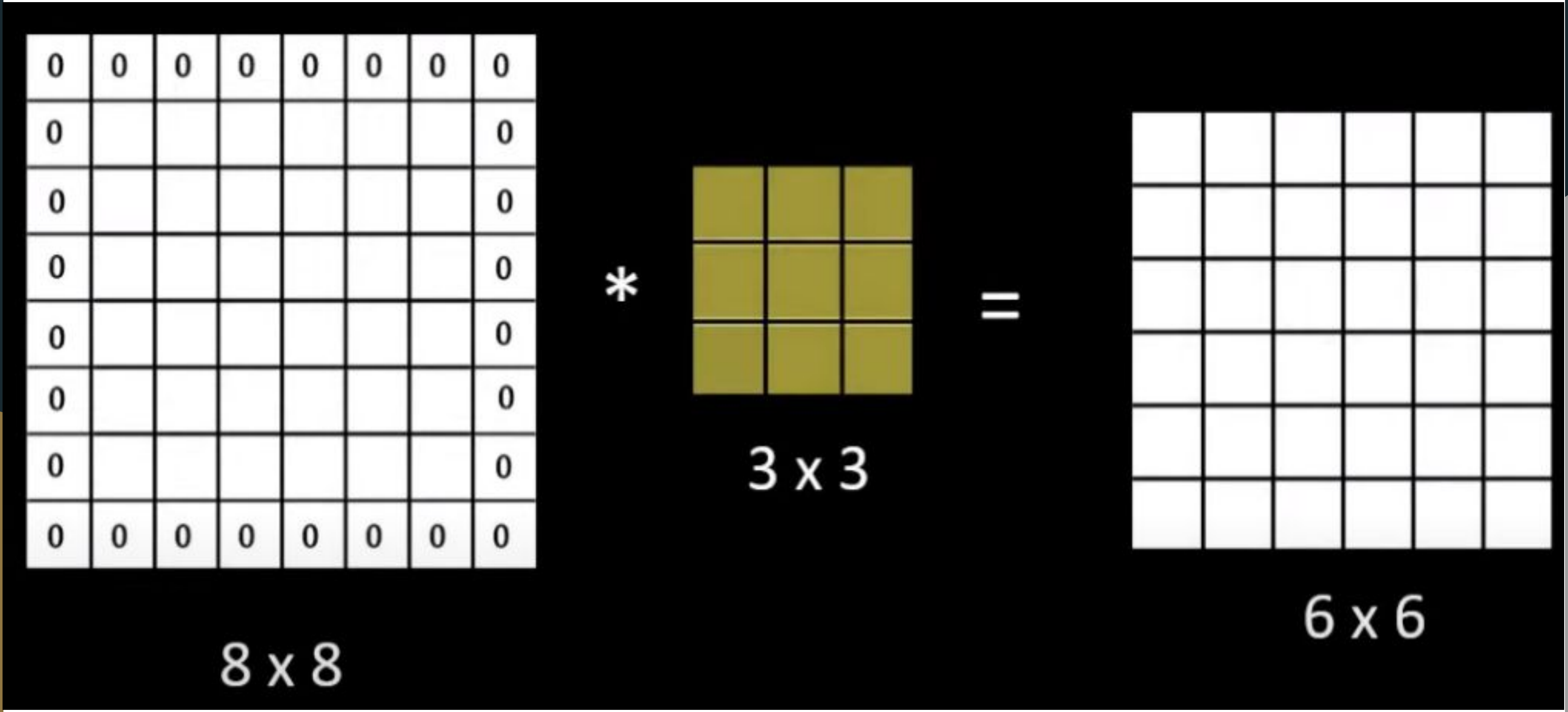

A Typical Convolutional Neural Network (CNN)
Convolution
Pooling
Convolution
Pooling
Kernel
Input Image
Featured maps
Pooled Featured maps
Featured maps
Pooled Featured maps
Flatten layer
Output
0.2
0.1
0.7
Donald
Goofy
Tweety
Feature Maps
Fully connected layer
Feature Extraction
Classification
Probabilistic distribution

## Padding

- In padding, one adds some “pixels” all around the borders of the feature map in order to maintain the spatial footprint.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | | | | | | | 0 |
| 0 | | | | | | | 0 |
| 0 | | | | | | | 0 |
| 0 | | | | | | | 0 |
| 0 | | | | | | | 0 |
| 0 | | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

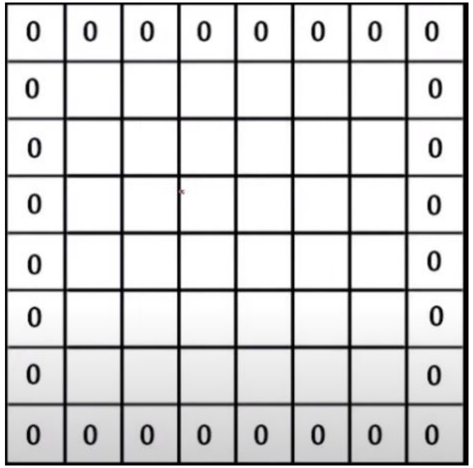


8 x 8
*
3 x 3
=
6 x 6

# Padding

- In padding, one adds some “pixels” all around the borders of the feature map in order to maintain the spatial footprint.
- n’=n+2p

  Same Convolution
- (n’-f+1)=n
- (n+2p-f+1)=n
- p=(f-1)/2



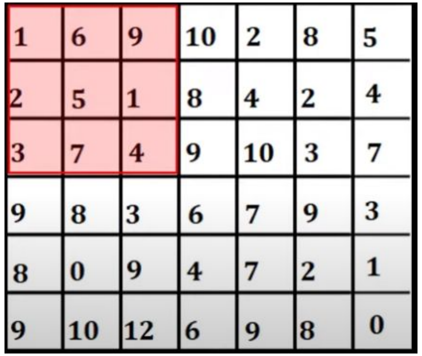
## Padding

- When padding is not used, the resulting “padding” is also referred to as a valid padding.
- Valid padding generally does not work well from an experimental point of view.
- Using half-padding ensures that some of the critical information at the borders of the layer is represented in a standalone way.
- In the case of valid padding, the contributions of the pixels on the borders of the layer will be under-represented compared to the central pixels in the next hidden layer, which is undesirable.
- Furthermore, this under-representation will be compounded over multiple layers.
- Therefore, padding is typically performed in all layers, and not just in the first layer where the spatial locations correspond to input values

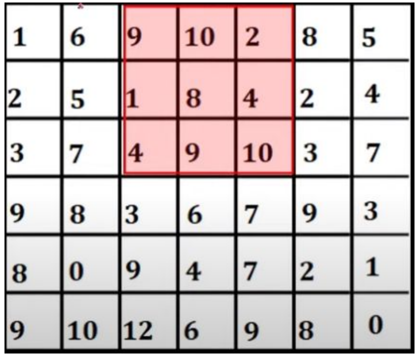
## Strides

- There are other ways in which convolution can reduce the spatial footprint of the image (or hidden layer).
- The above approach performs the convolution at every position in the spatial location of the feature map.
- However, it is not necessary to perform the convolution at every spatial position in the layer.
- One can reduce the level of granularity of the convolution by using the notion of strides.
- The description above corresponds to the case when a stride of 1 is used.

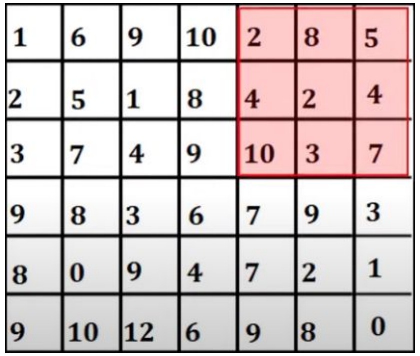
# Strides

- Stride=2
- 

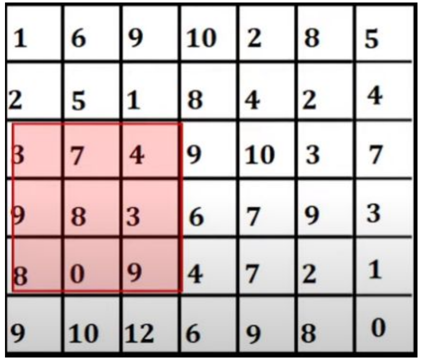
|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 6 | 9 | 10 | 2 | 8 | 5 |
| 2 | 5 | 1 | 8 | 4 | 2 | 4 |
| 3 | 7 | 4 | 9 | 10 | 3 | 7 |
| 9 | 8 | 3 | 6 | 7 | 9 | 3 |
| 8 | 0 | 9 | 4 | 7 | 2 | 1 |
| 9 | 10 | 12 | 6 | 9 | 8 | 0 |



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



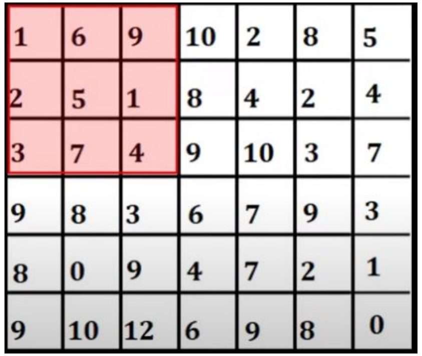
|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 6 | 9 | 10 | 2 | 8 | 5 |
| 2 | 5 | 1 | 8 | 4 | 2 | 4 |
| 3 | 7 | 4 | 9 | 10 | 3 | 7 |
| 9 | 8 | 3 | 6 | 7 | 9 | 3 |
| 8 | 0 | 9 | 4 | 7 | 2 | 1 |
| 9 | 10 | 12 | 6 | 9 | 8 | 0 |



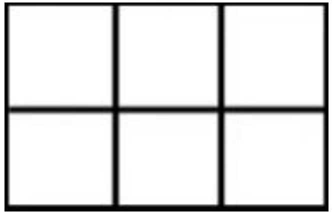
|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 6 | 9 | 10 | 2 | 8 | 5 |
| 2 | 5 | 1 | 8 | 4 | 2 | 4 |
| 3 | 7 | 4 | 9 | 10 | 3 | 7 |
| 9 | 8 | 3 | 6 | 7 | 9 | 3 |
| 8 | 0 | 9 | 4 | 7 | 2 | 1 |
| 9 | 10 | 12 | 6 | 9 | 8 | 0 |

# Strides

- Stride=2

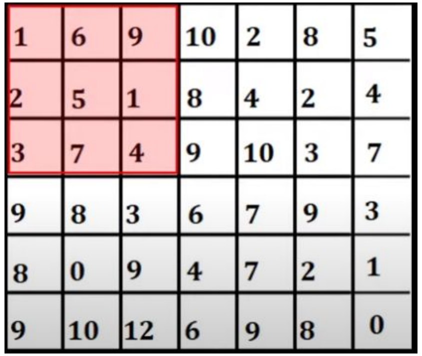


| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 6 | 9 | 10 | 2 | 8 | 5 |
| 2 | 5 | 1 | 8 | 4 | 2 | 4 |
| 3 | 7 | 4 | 9 | 10 | 3 | 7 |
| 9 | 8 | 3 | 6 | 7 | 9 | 3 |
| 8 | 0 | 9 | 4 | 7 | 2 | 1 |
| 9 | 10 | 12 | 6 | 9 | 8 | 0 |

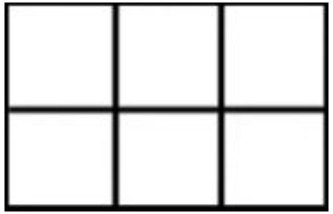


# Strides

- Stride=2



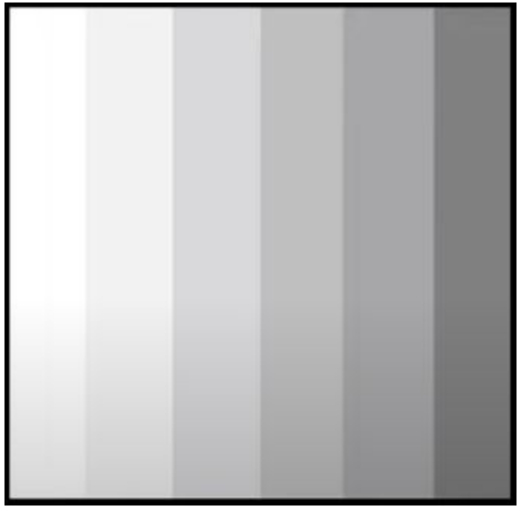
| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 6 | 9 | 10 | 2 | 8 | 5 |
| 2 | 5 | 1 | 8 | 4 | 2 | 4 |
| 3 | 7 | 4 | 9 | 10 | 3 | 7 |
| 9 | 8 | 3 | 6 | 7 | 9 | 3 |
| 8 | 0 | 9 | 4 | 7 | 2 | 1 |
| 9 | 10 | 12 | 6 | 9 | 8 | 0 |

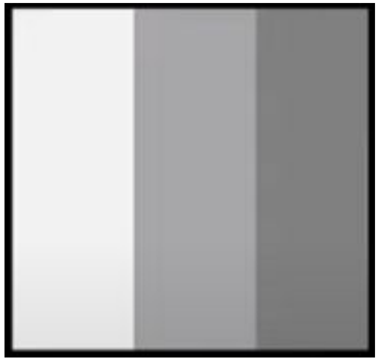


$$[(n-f)/s + 1]$$

## Max Pooling

- The pooling operation works on small grid regions of size Pq × Pq in each layer, and produces another layer with the same depth (unlike filters).
- For each square region of size Pq × Pq in each of the dq activation maps, the maximum of these values is returned. This approach is referred to as max-pooling



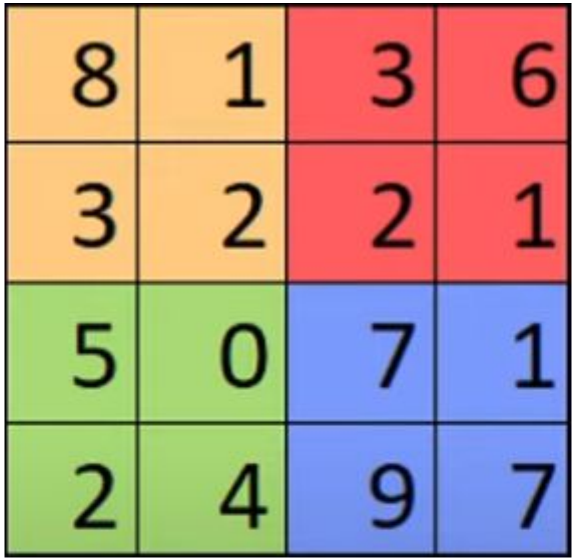


## Max Pooling

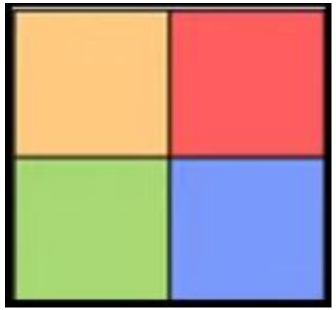
- Unlike with convolution operations, pooling is done at the level of each activation map.
- Whereas a convolution operation simultaneously uses all dq feature maps in combination with a filter to produce a single feature value, pooling independently operates on each feature map to produce another feature map.
- Therefore, the operation of pooling does not change the number of feature maps.

## Max Pooling

- Fix Filter size and a stride size

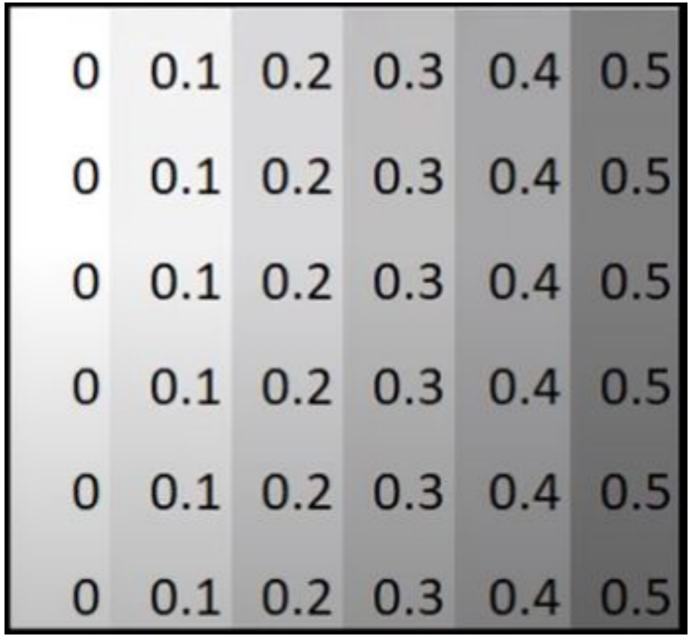


| | | | |
|---|---|---|---|
| 8 | 1 | 3 | 6 |
| 3 | 2 | 2 | 1 |
| 5 | 0 | 7 | 1 |
| 2 | 4 | 9 | 7 |

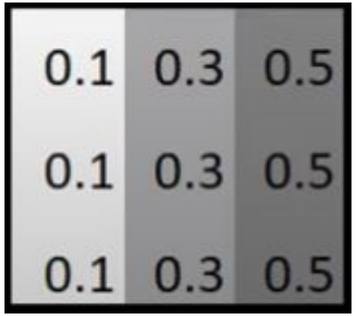


## Max Pooling

- It reduces the image size  and hence reduce the computational cost.



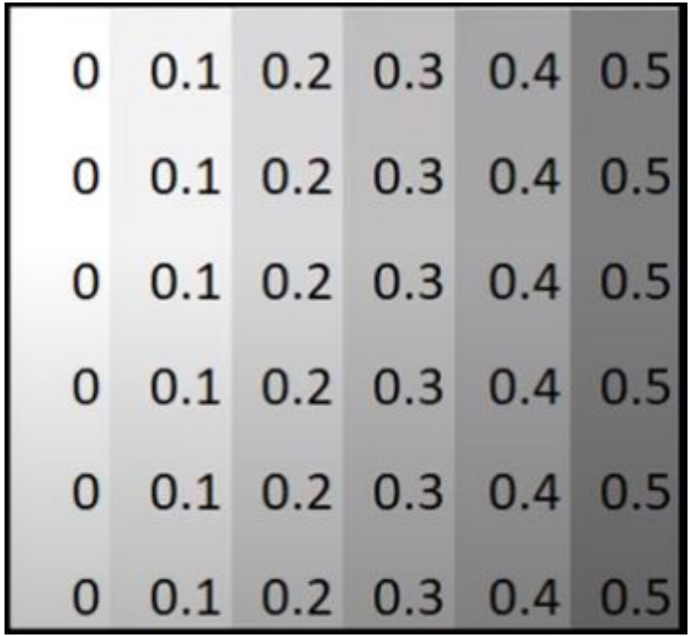
| | | | | | |
|---|---|---|---|---|---|
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |



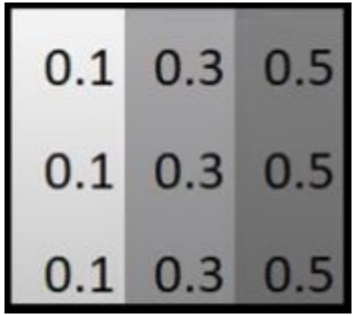
| | | |
|---|---|---|
| 0.1 | 0.3 | 0.5 |
| 0.1 | 0.3 | 0.5 |
| 0.1 | 0.3 | 0.5 |

# Max Pooling

- It reduces the image size  and hence reduce the computational cost.



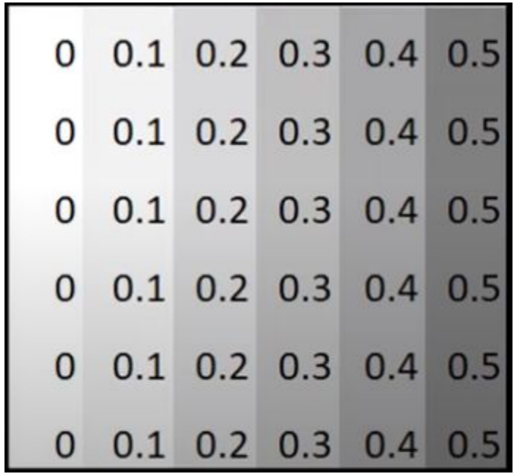
| | | | | | |
|---|---|---|---|---|---|
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |



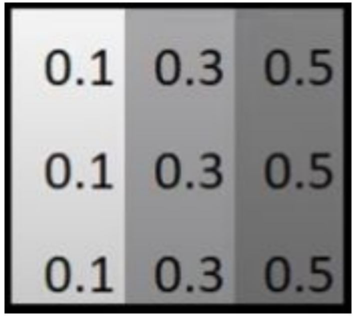
| | | |
|---|---|---|
| 0.1 | 0.3 | 0.5 |
| 0.1 | 0.3 | 0.5 |
| 0.1 | 0.3 | 0.5 |

## Max Pooling

- It reduces the image size  and hence reduce the computational cost.
- Sharpens/ enhances features.



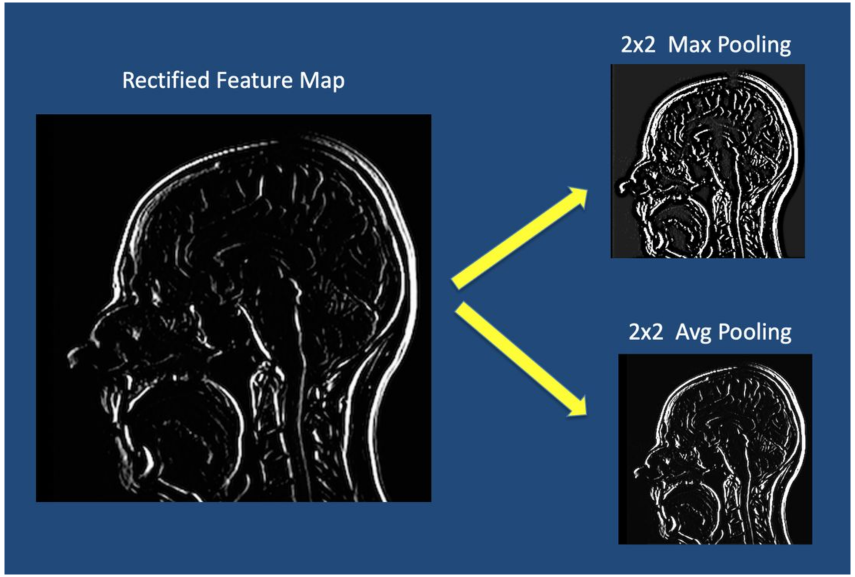
| | | | | | |
|---|---|---|---|---|---|
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |

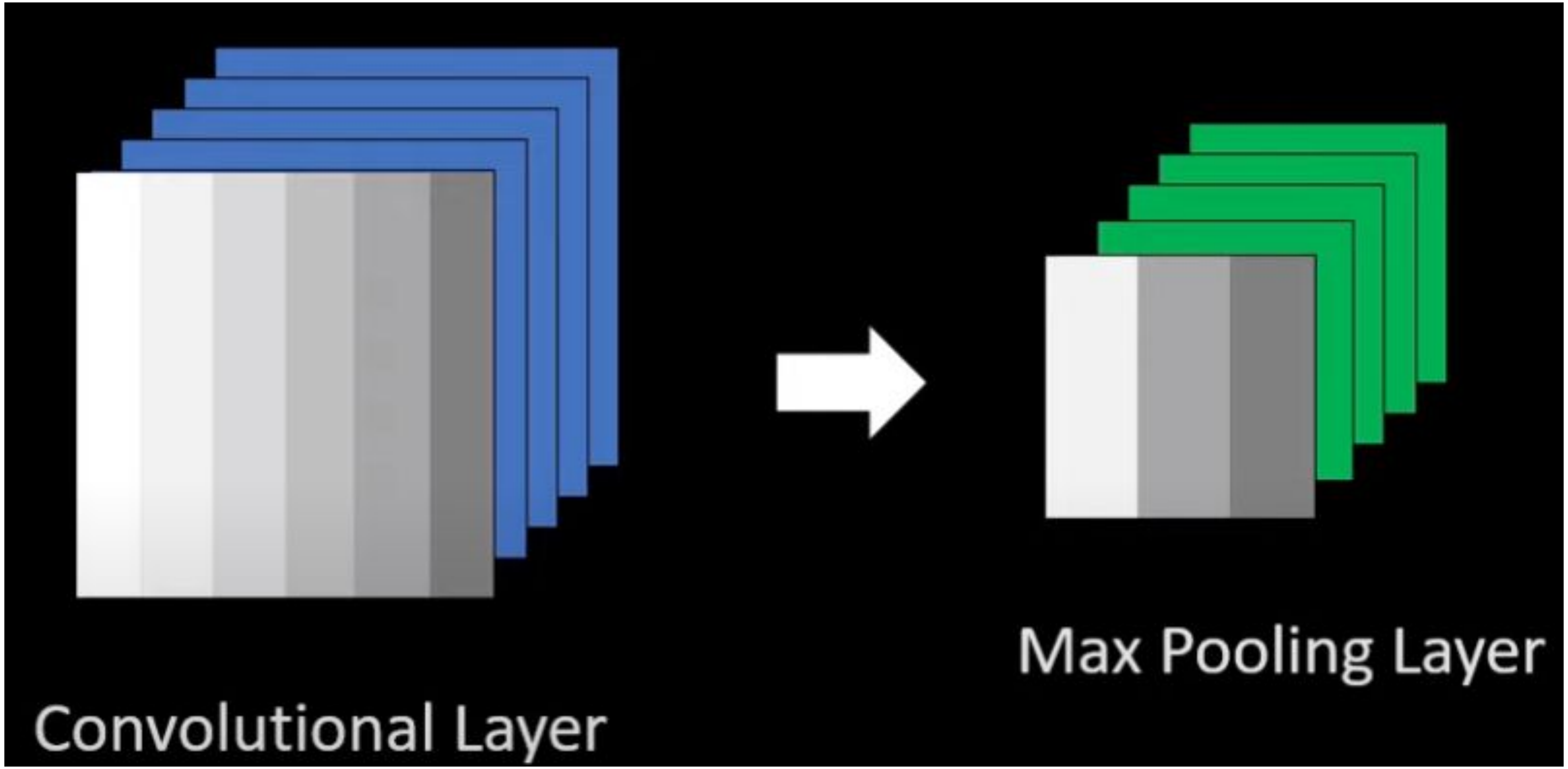


| | | |
|---|---|---|
| 0.1 | 0.3 | 0.5 |
| 0.1 | 0.3 | 0.5 |
| 0.1 | 0.3 | 0.5 |

## Max Pooling

- It reduces the image size  and hence reduce the computational cost.
- Sharpens/ enhances features.
- It is applied after convolution layer.




Convolutional Layer
Max Pooling Layer