Introduction to Machine Learning 4:

Convolutional Neural Networks

Spot the Difference

(Neamblysomus julianae)

Department of Biology
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway

Male Bullock's Oriole; by Kevin Cole, WikiMedia.Org, Creative Commons 2.0

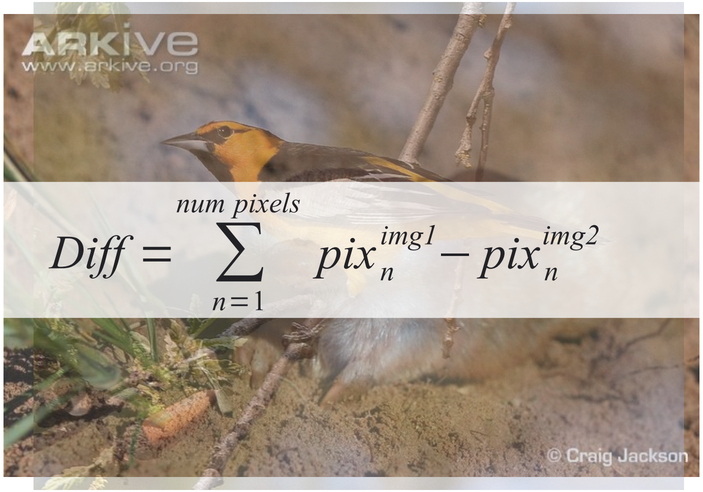
(Neamblysomus julianae)

© Craig Jackson
Department of Biology
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway

Take a moment to appreciate the cuteness

Spot the Difference

Pixel by Pixel
comparison?



$$Diff = \sum_{n=1}^{num\ pixels} pix_n^{img1} - pix_n^{img2}$$

L1
distance

Spot the Difference

Pixel by Pixel comparison?

$$Diff = \sum_{n=1}^{num\ pixels} pix_n^{img\,1} - pix_n^{img\,2}$$

L1 distance

Compare each image to all others, calculate distances,

Then use a clustering algorithm to perform classification: N-nearest neighbours

Spot the Difference

Pixel by Pixel comparison / Nearest Neighbour

As you might have guessed this, doesn't really work that well...

Spot the Difference

Problems:


ARKive
www.arkive.org
© Craig Jackson

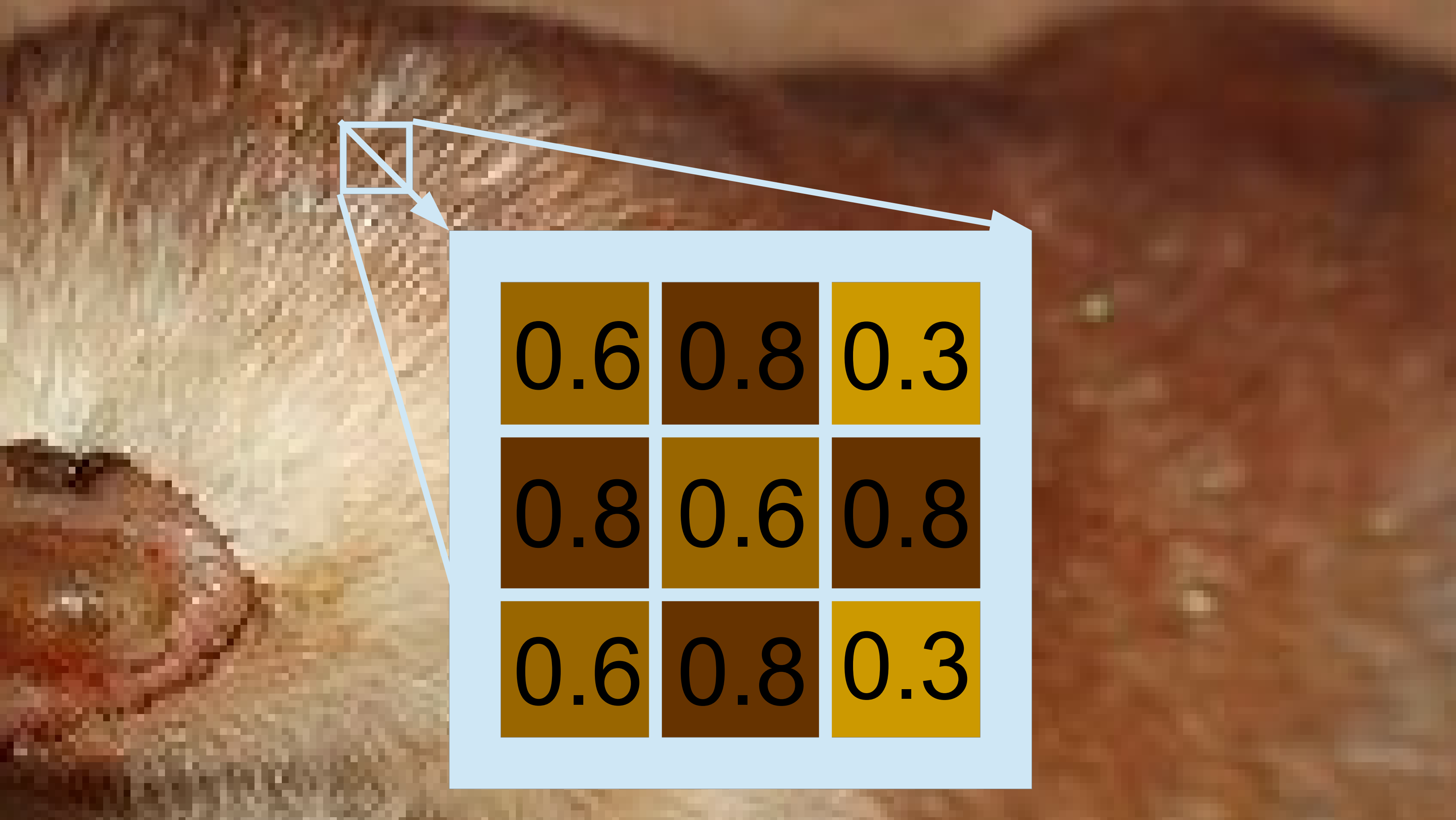
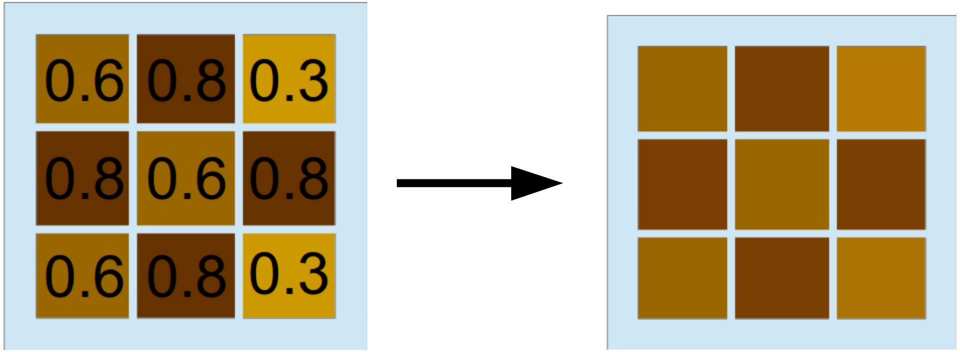

0.6 0.8 0.3
0.8 0.6 0.8
0.6 0.8 0.3

Image processing pixel weighting

We can perform image processing by looking at groups of pixels in an image and summing them up in different ways

For example we can cause a blur effect by averaging the values of neighboring pixels



Grey Value Pixels in Image

0.5	0.3	0.8	0.3
0.3	0.8	0.5	0.3
0.5	0.8	0.3	0.2
0.8	0.3	0.2	0.2

Grey Value Pixels in Image

0.5	0.3	0.8	0.3
0.3	0.8	0.5	0.3
0.5	0.8	0.3	0.2
0.8	0.3	0.2	0.2

We want to detect Patterns!

- Lines
- Edges
- Shapes
- Curves
- Etc...

Simplified Image

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

Simplified Image

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

“Kernel” or “Filter”

1	0	1
0	1	0
1	0	1

Simplified Image

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

“Kernel” or “Filter”

1	0	1
0	1	0
1	0	1

Simplified Image

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

The Kernel ‘Slides’ over the image multiplying the values of the image with the values in the Kernel.
This is simple Matrix element-wise multiplication.

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Simplified Image

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

The Kernel ‘Slides’ over the image multiplying the values of the image with the values in the Kernel.
This is simple Matrix element-wise multiplication.

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Simplified Image

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

The Kernel ‘Slides’ over the image multiplying the values of the image with the values in the Kernel.
This is simple Matrix element-wise multiplication.

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Simplified Image

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

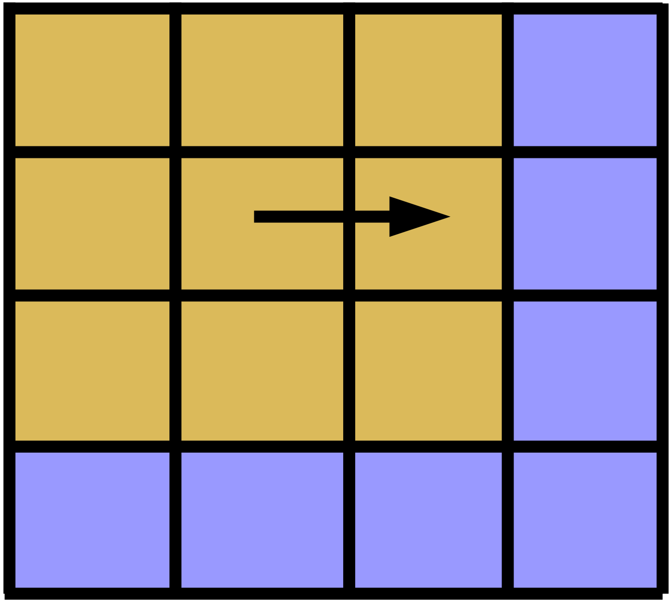
The Kernel ‘Slides’ over the image multiplying the values of the image with the values in the Kernel.
This is simple Matrix element-wise multiplication.

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Convolution Demo

Simplified Image

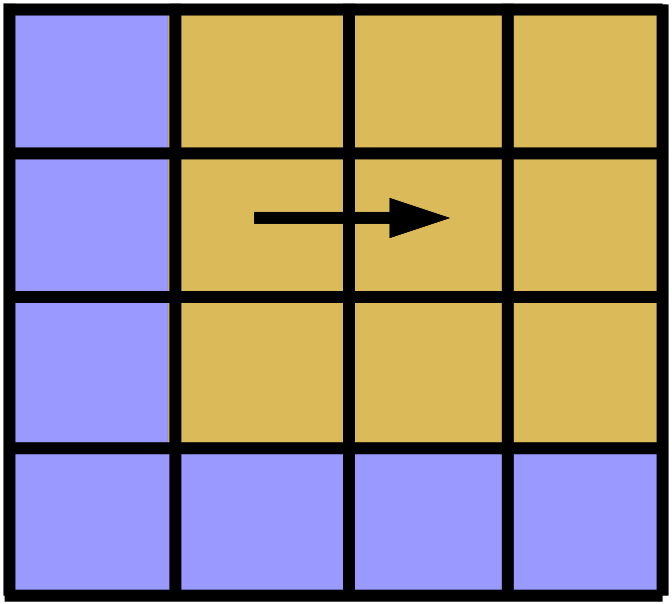


The ‘Stride’ is the amount of pixels the Kernel moves at each step, in this case, the Stride = 1

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Simplified Image

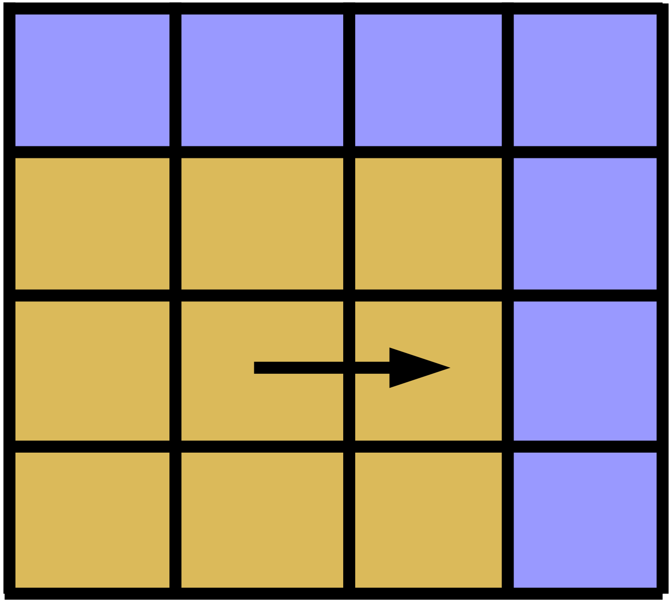


The ‘Stride’ is the amount of pixels the Kernel moves at each step, in this case, the Stride = 1

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Simplified Image

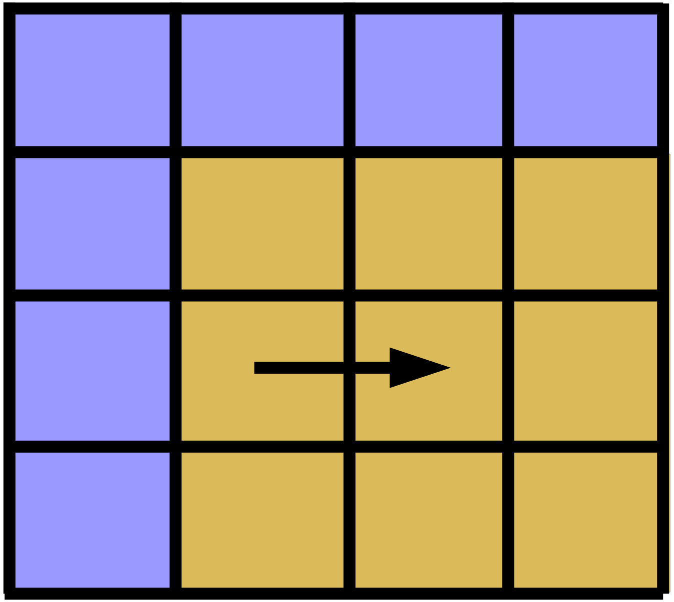


The ‘Stride’ is the amount of pixels the Kernel moves at each step, in this case, the Stride = 1

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Simplified Image



The 'Stride' is the amount of pixels the Kernel moves at each step, in this case, the Stride = 1

"Kernel" or "Filter"

1	0	1
0	1	0
1	0	1

Stride

- Can be changed depending on the size of the image or the patterns being looked for.

  A large image might not need every pixel examined, since patterns might extend well beyond that scale. Kernel size can also play a role.

Simplified Image

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

Kernel creates a new matrix, or convolved image

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

Image

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

“Kernel”
or
“Filter”

1	0	1
0	1	0
1	0	1

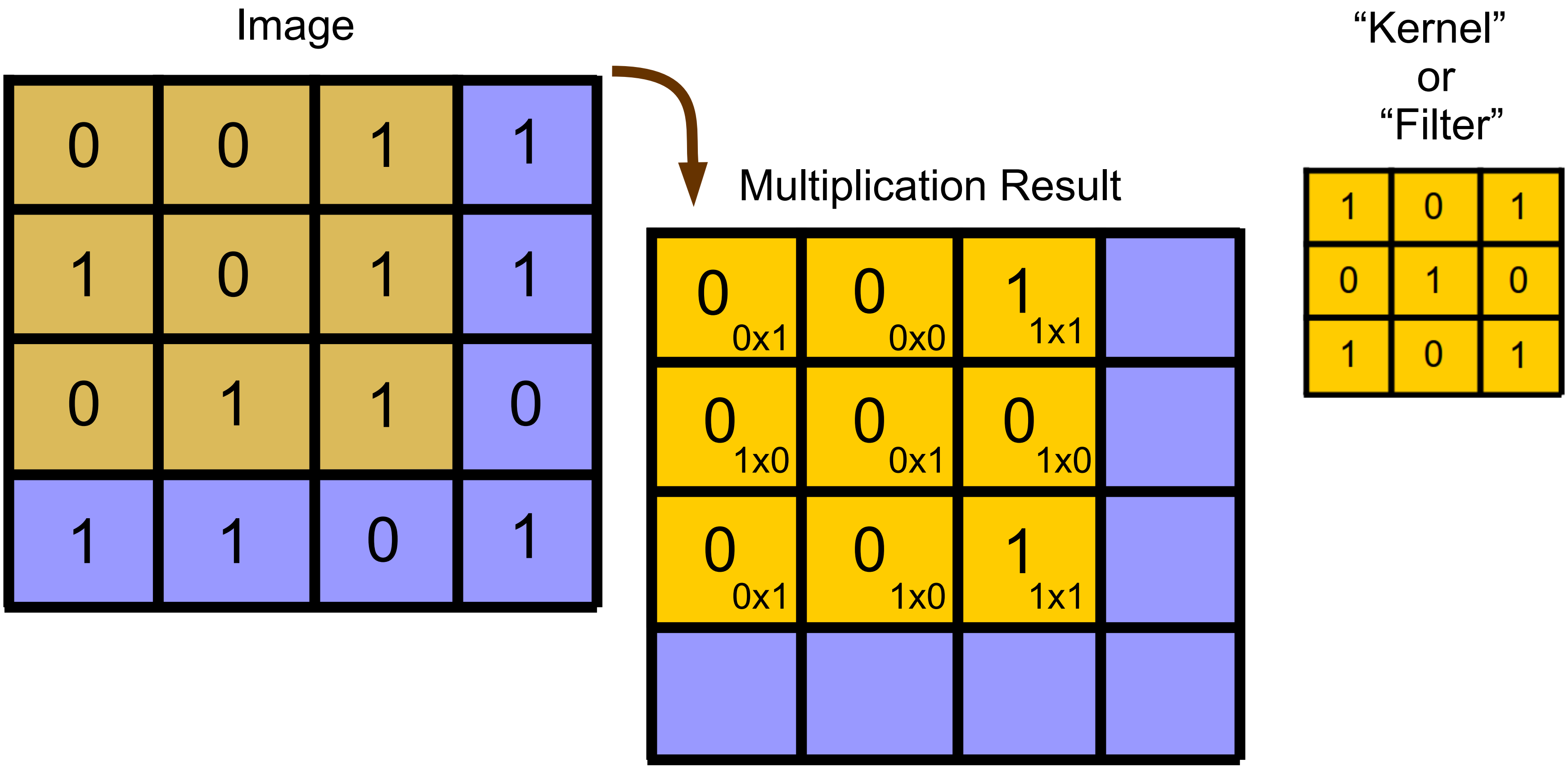


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
0 0x1
0 0x0
1 1x1
0 1x0
0 0x1
0 1x0
0 0x1
0 1x0
1 1x1
“Kernel”
or
“Filter”
1 0 1
0 1 0
1 0 1

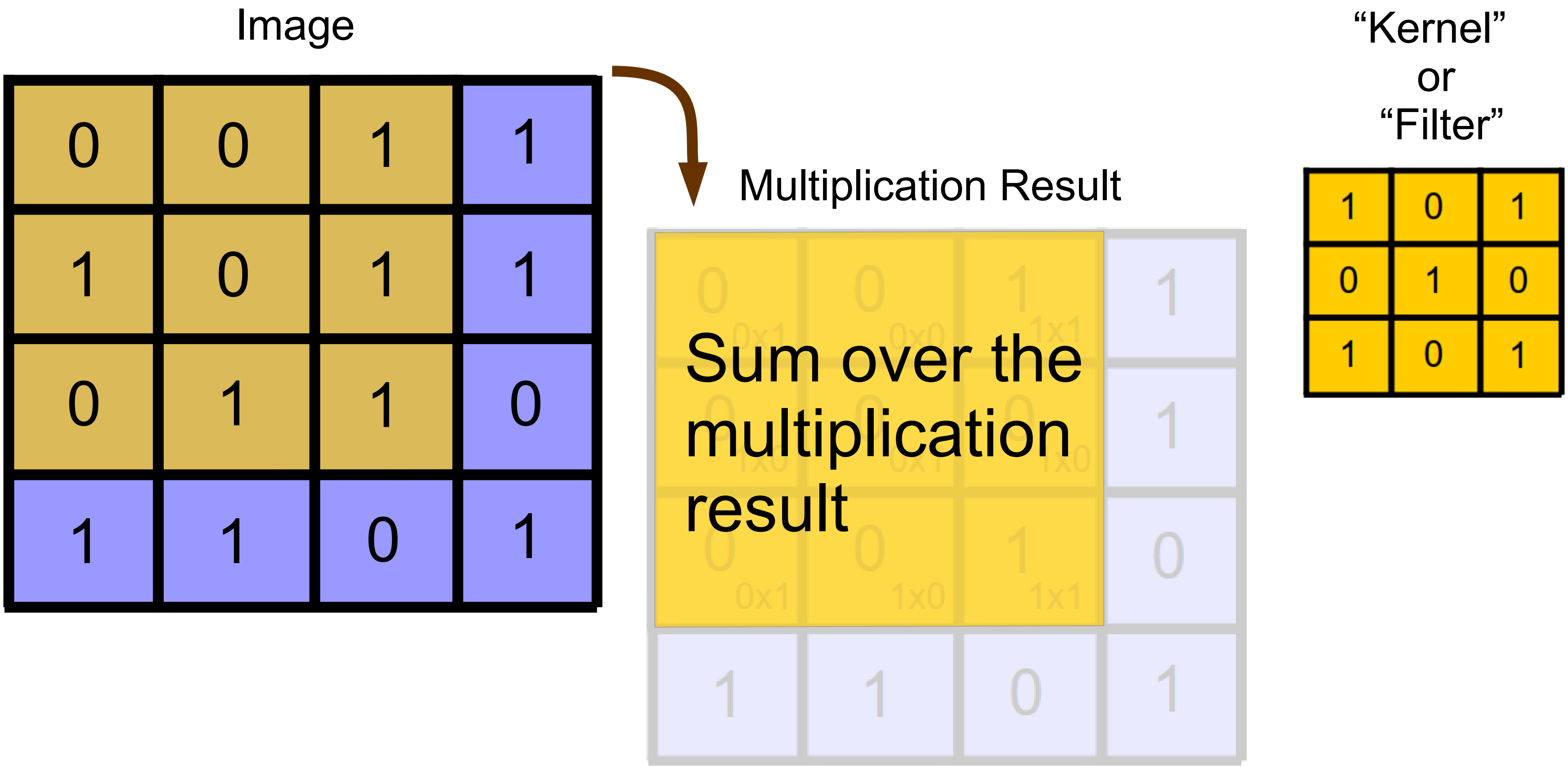


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
Sum over the multiplication result
“Kernel” or “Filter”
1 0 1
0 1 0
1 0 1

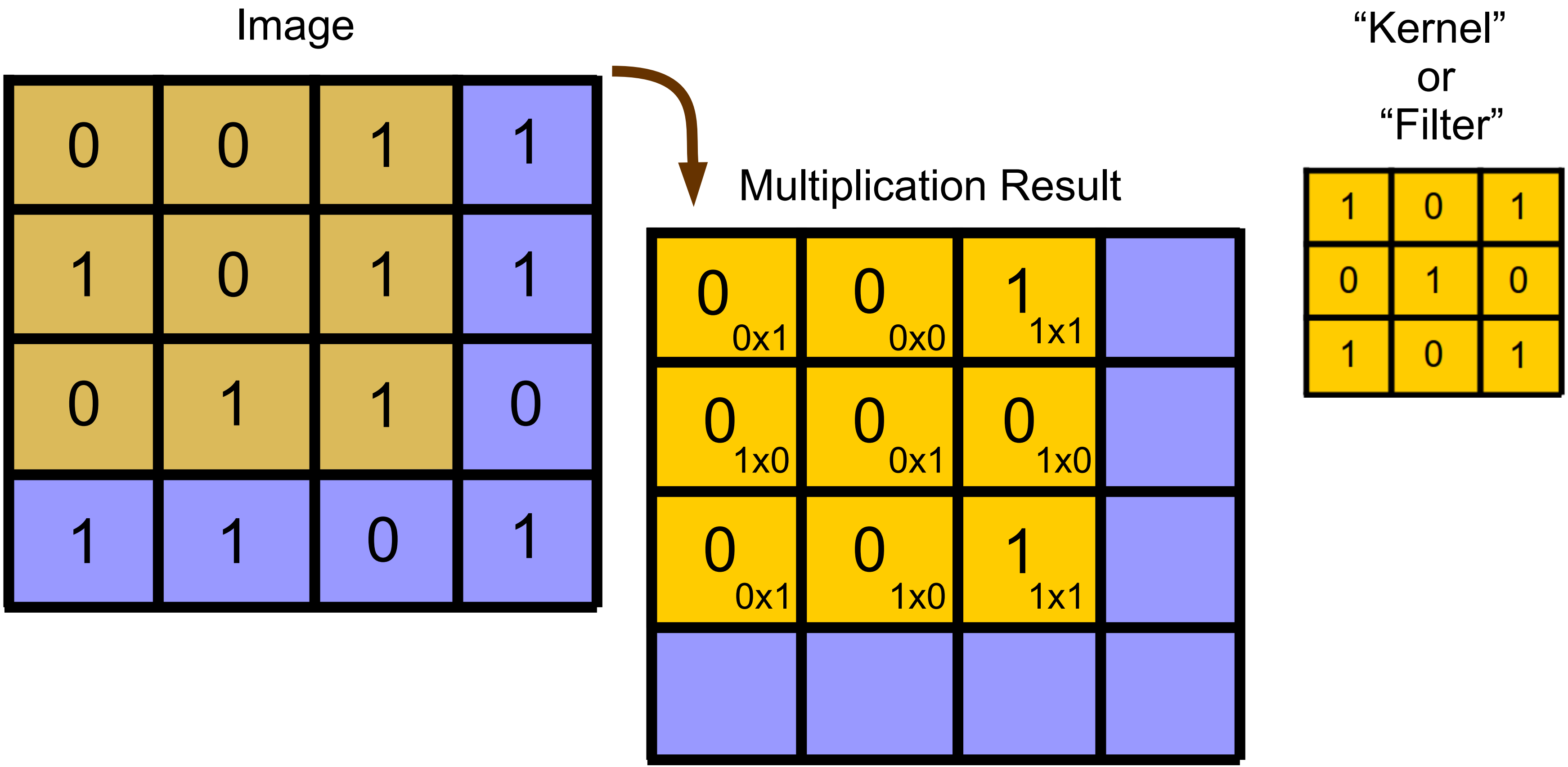


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
0 0x1
0 0x0
1 1x1
0 1x0
0 0x1
0 1x0
0 0x1
0 1x0
1 1x1
“Kernel” or “Filter”
1 0 1
0 1 0
1 0 1

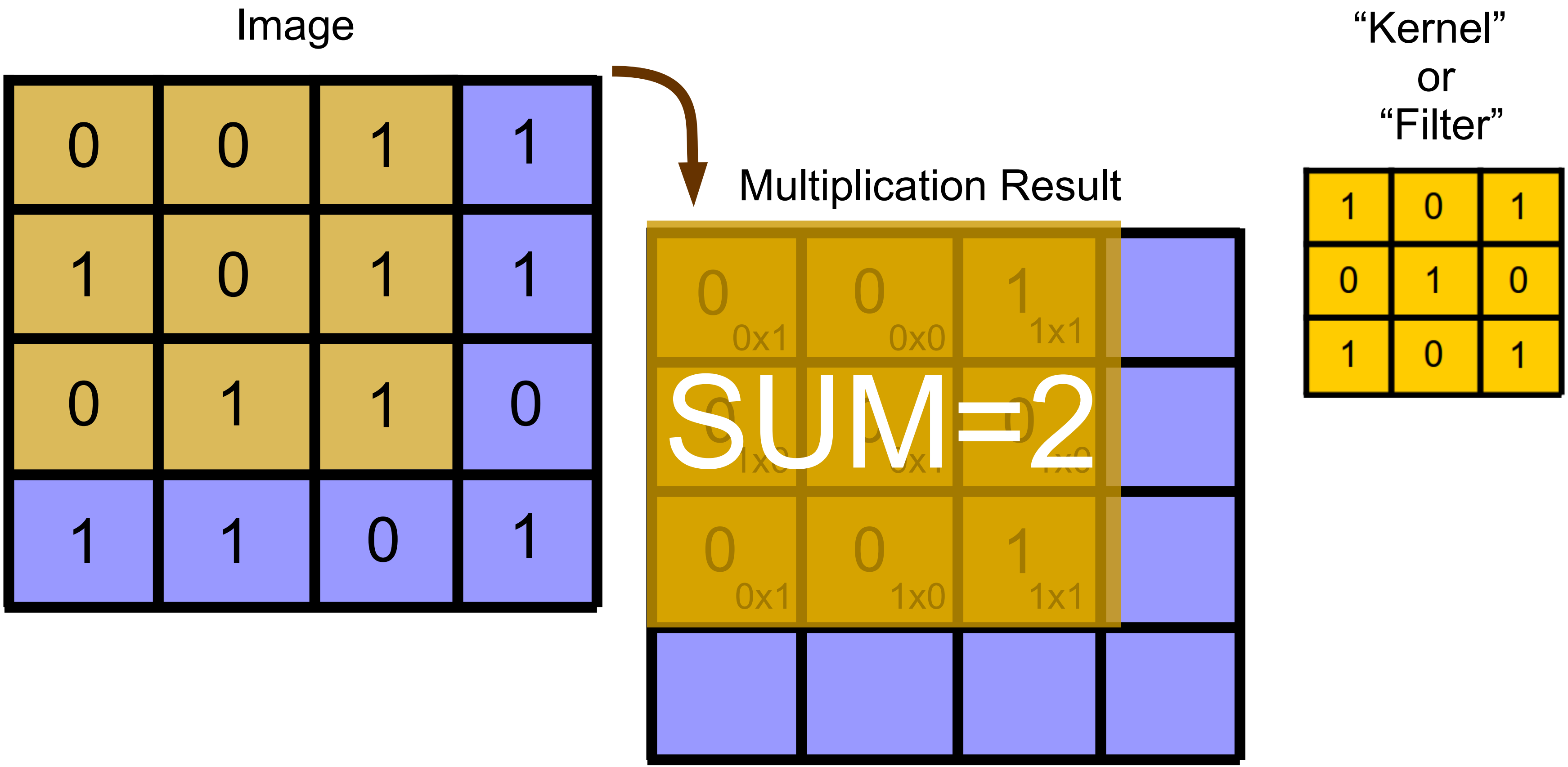


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
0 0x1
0 0x0
1 1x1
SUM=2
0 0x1
0 1x0
1 1x1
“Kernel”
or
“Filter”
1 0 1
0 1 0
1 0 1

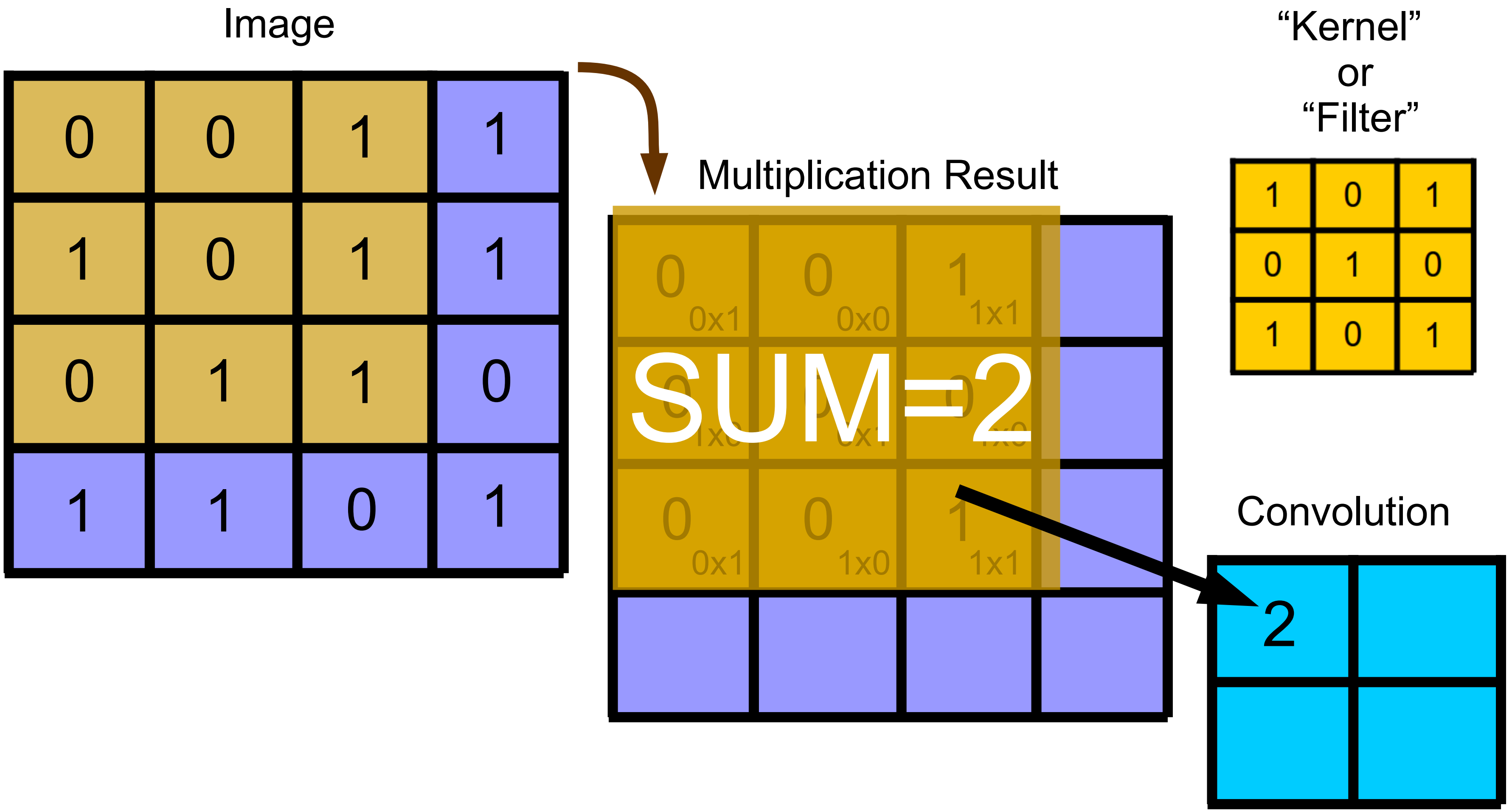


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
0 0x1
0 0x0
1 1x1
0 1x0
0 0x1
0 1x0
0 0x1
0 1x0
1 1x1
SUM=2
"Kernel"
or
"Filter"
1 0 1
0 1 0
1 0 1
Convolution
2

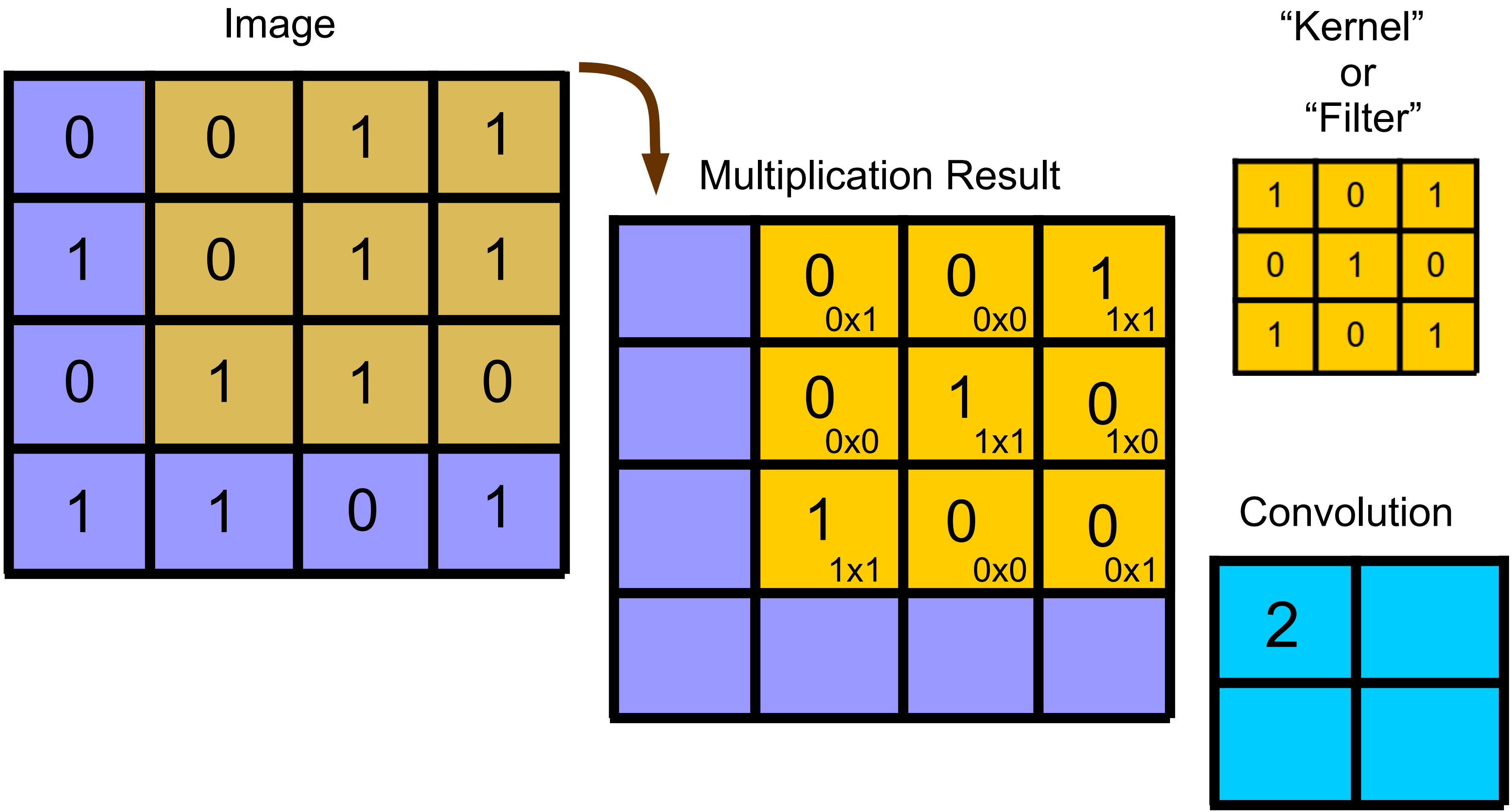


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
0 0x1
0 0x0
1 1x1
0 0x0
1 1x1
0 1x0
1 1x1
0 0x0
0 0x1
“Kernel” or “Filter”
1 0 1
0 1 0
1 0 1
Convolution
2

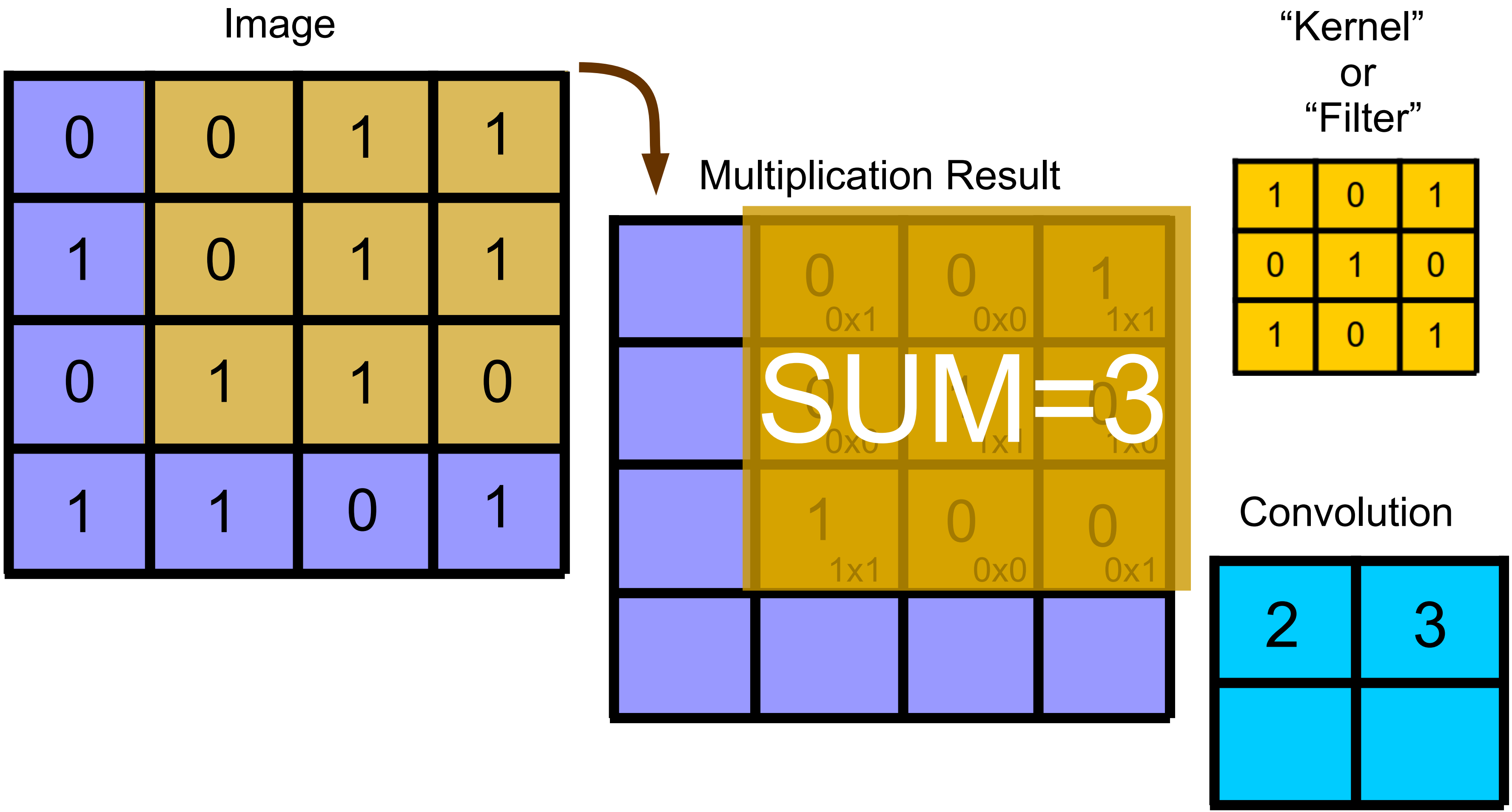


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
0 0x1
0 0x0
1 1x1
SUM=3
1 1x1
0 0x0
0 0x1
“Kernel”
or
“Filter”
1 0 1
0 1 0
1 0 1
Convolution
2 3

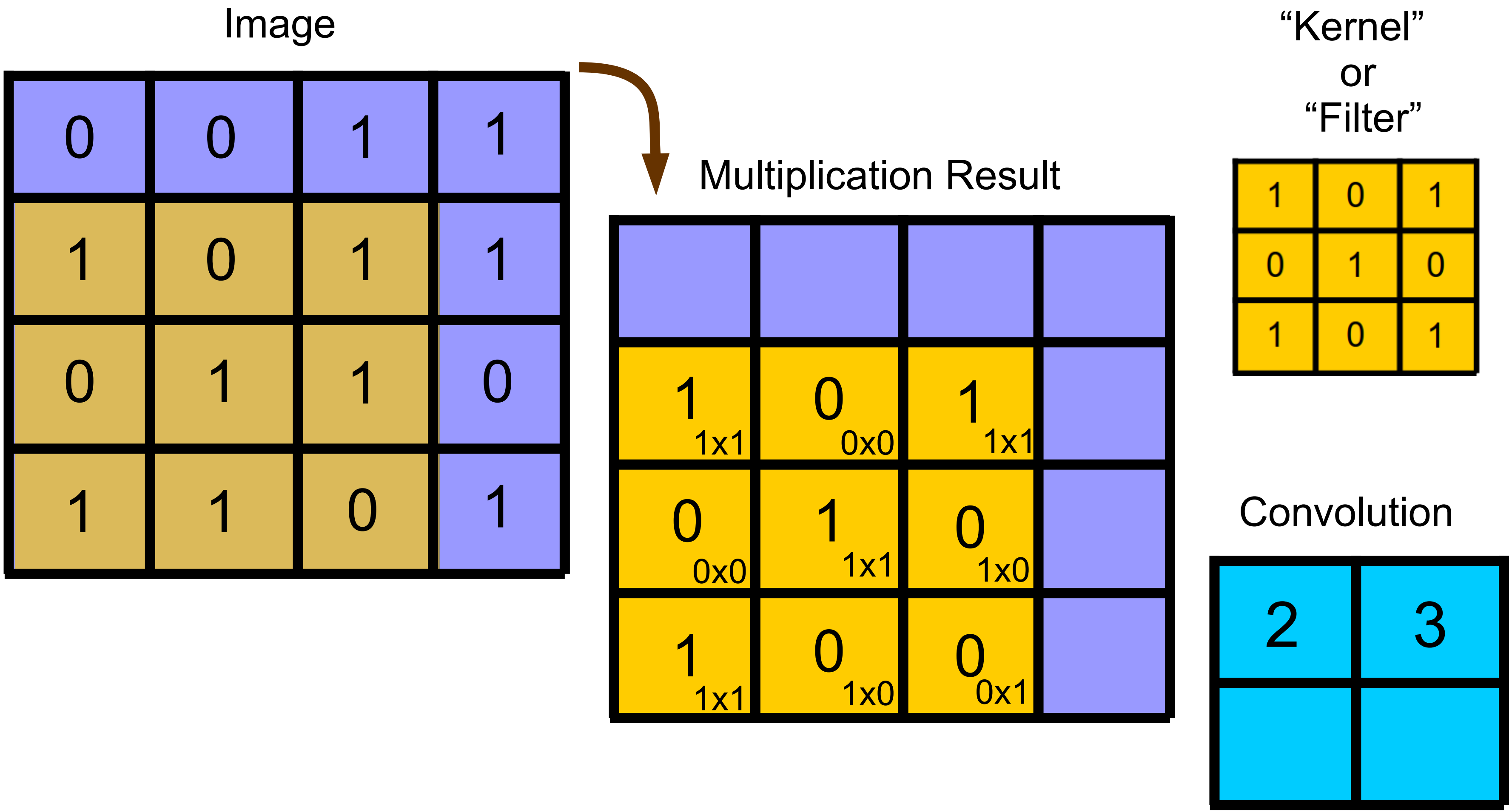


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
1 1x1
0 0x0
1 1x1
0 0x0
1 1x1
0 1x0
1 1x1
0 1x0
0 0x1
"Kernel"
or
"Filter"
1 0 1
0 1 0
1 0 1
Convolution
2 3

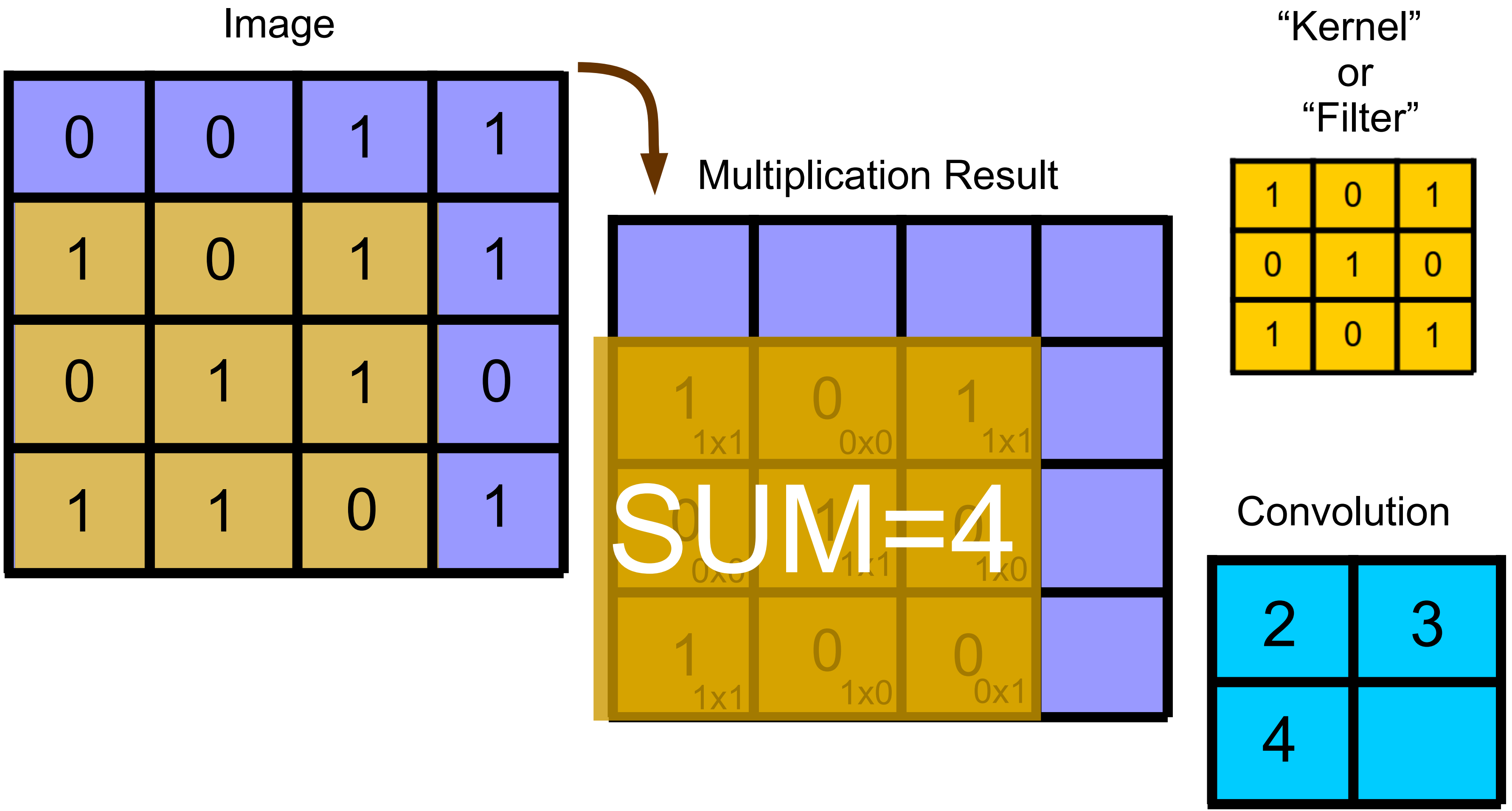


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
1 1x1
0 0x0
1 1x1
0 0x0
1 1x1
0 1x0
1 1x1
0 1x0
0 0x1
SUM=4
"Kernel"
or
"Filter"
1 0 1
0 1 0
1 0 1
Convolution
2 3
4

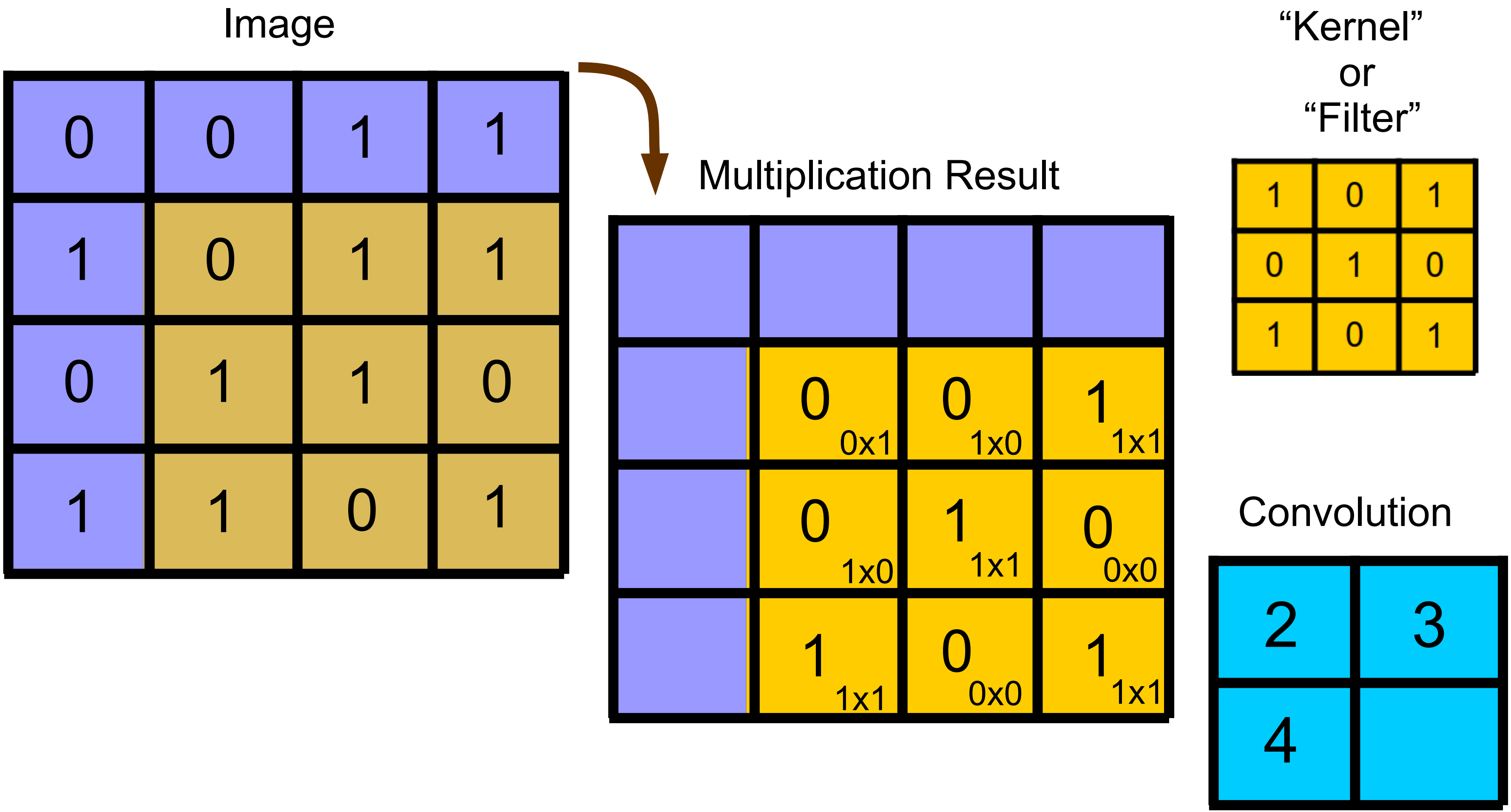


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
Multiplication Result
0 0x1
0 1x0
1 1x1
0 1x0
1 1x1
0 0x0
1 1x1
0 0x0
1 1x1
“Kernel”
or
“Filter”
1 0 1
0 1 0
1 0 1
Convolution
2 3
4

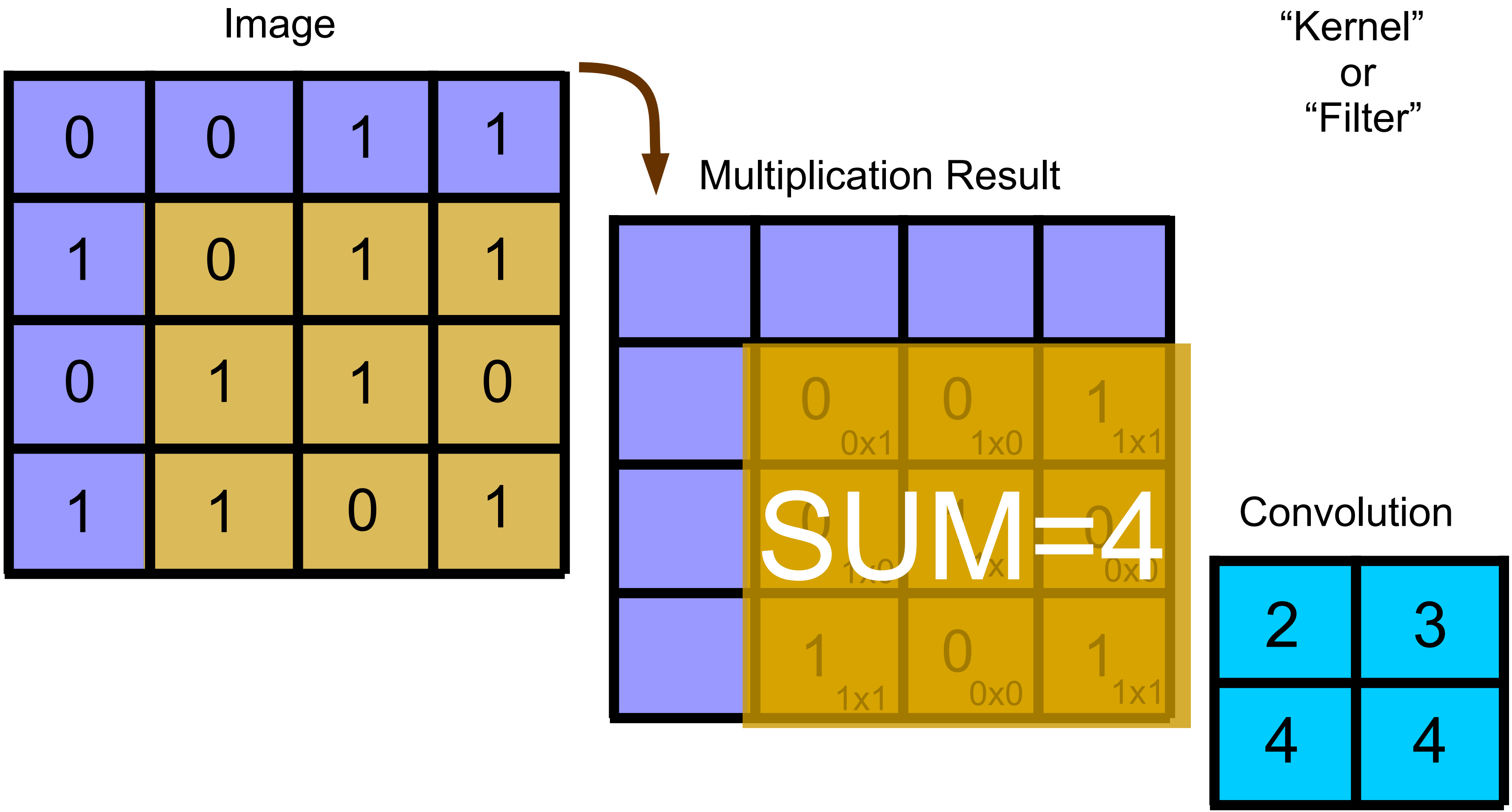


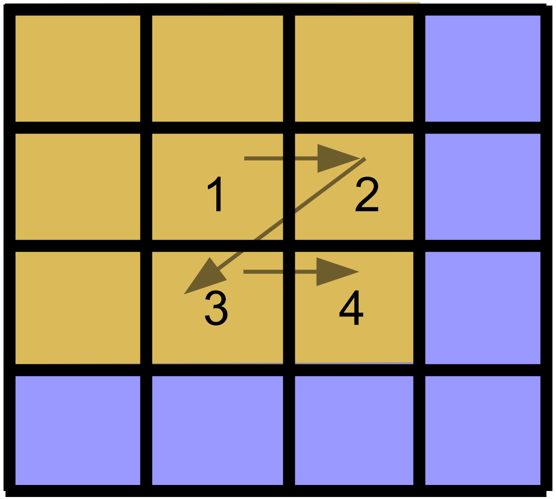
Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
“Kernel”
or
“Filter”
Multiplication Result
0 0x1
0 1x0
1 1x1
SUM=4
1 1x1
0 0x0
1 1x1
Convolution
2 3
4 4

Various Kernels Demo

Padding

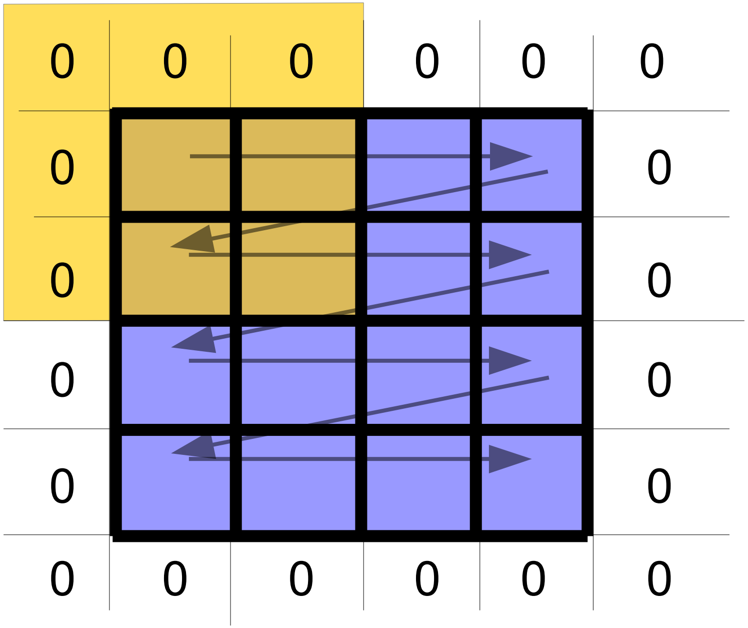
- Using the method we have shown, convolutions will always be smaller than the input image,
- To change this, we can

  use Zero-Padding



Padding

- Using the method we have shown, convolutions will always be smaller than the input image,
- To change this, we can use Zero-Padding



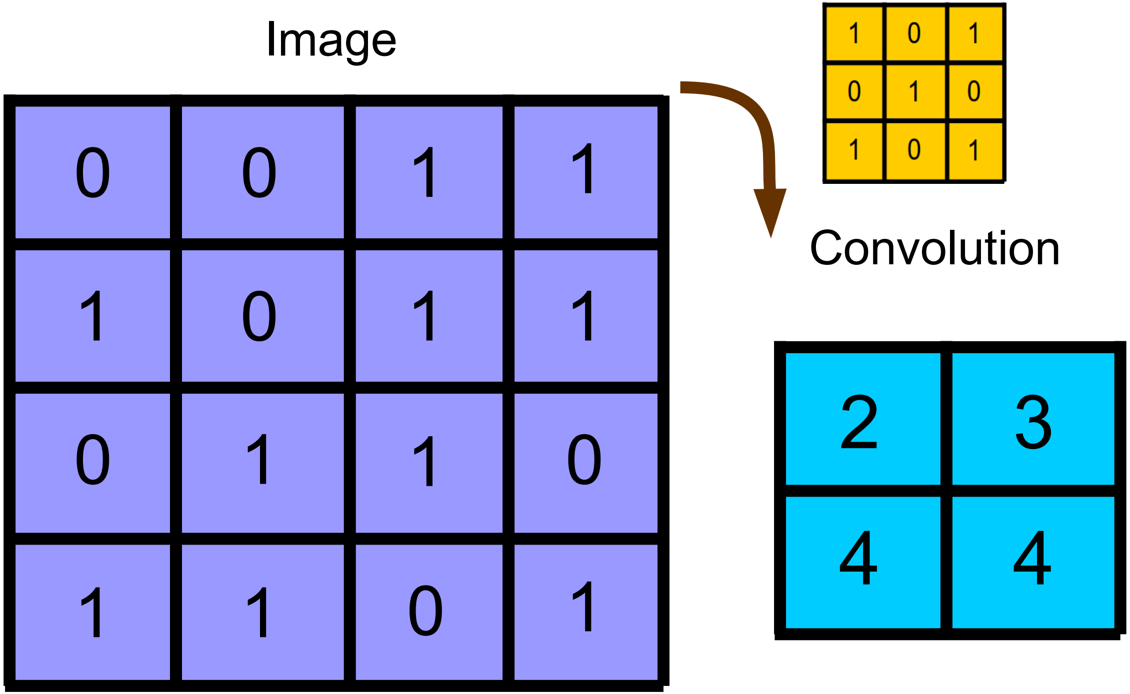
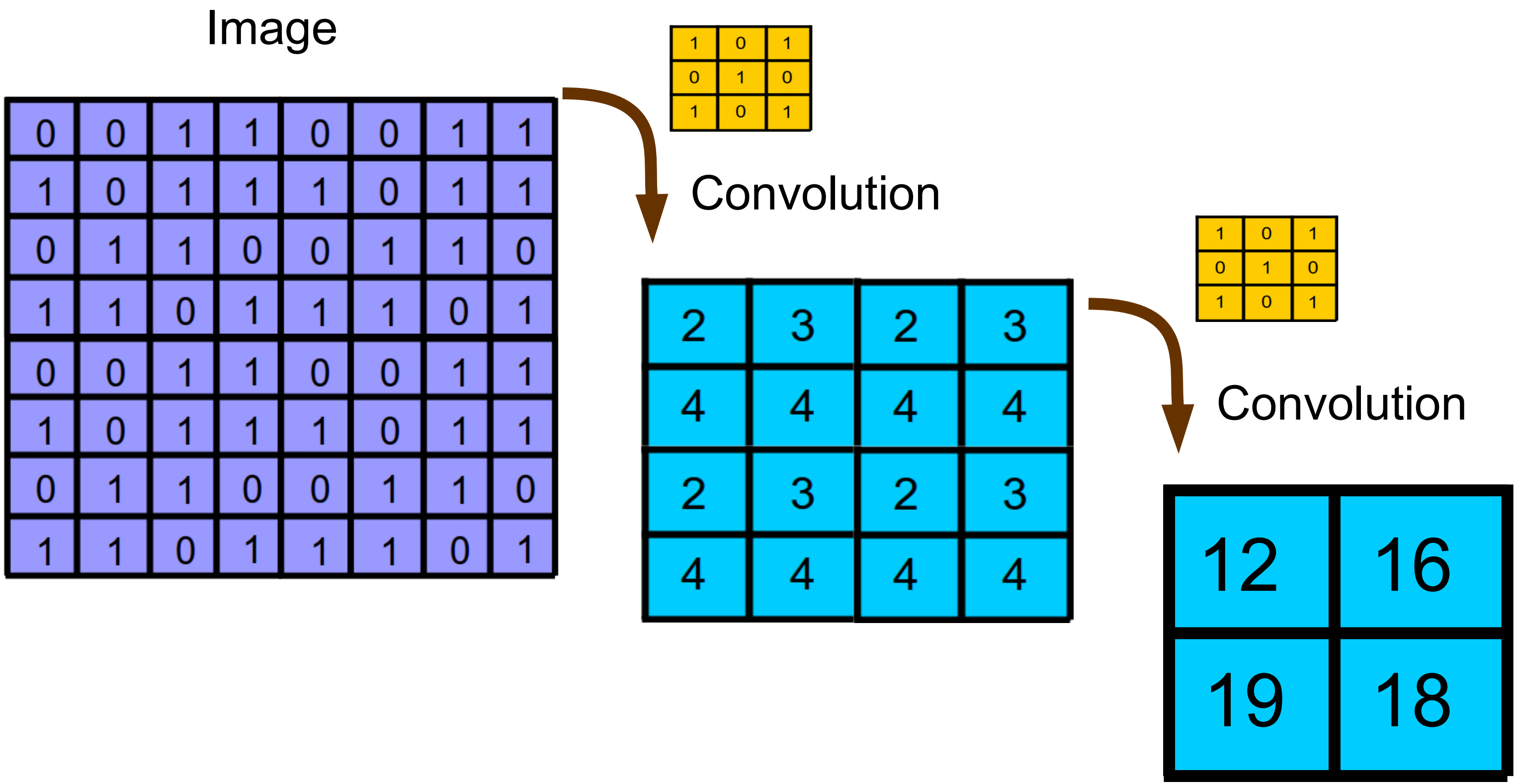


Image
0 0 1 1
1 0 1 1
0 1 1 0
1 1 0 1
1 0 1
0 1 0
1 0 1
Convolution
2 3
4 4

Convolutions Stack...

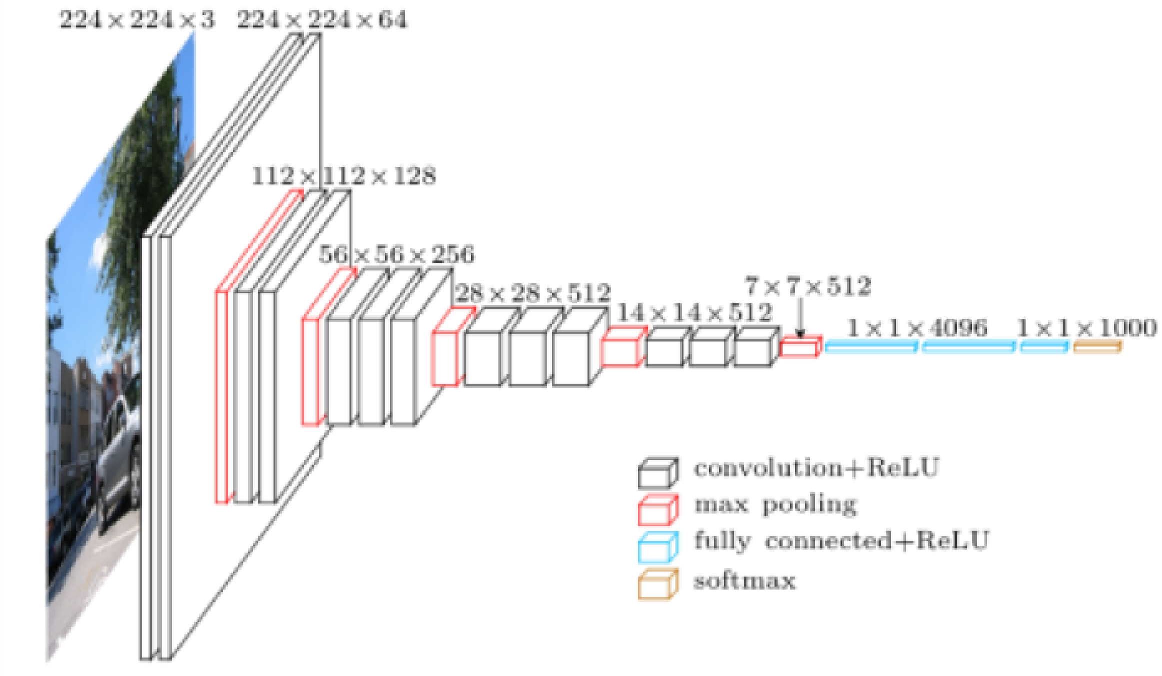


Kernels are ‘Trained’ along with the NN

- The values of the kernel matrix are adjusted along with the weights in the optimization process.

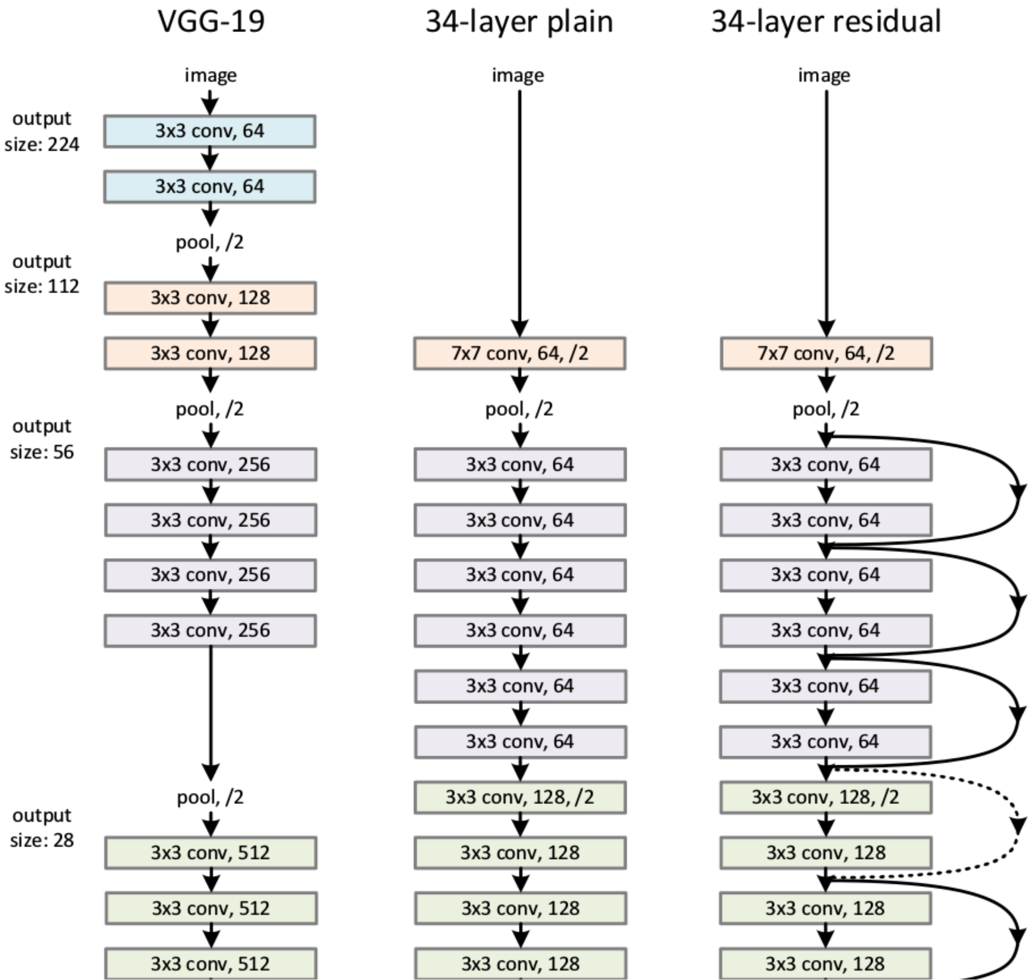
Visualizing Convolutional Networks filtered Images

Typical Architectures



VGG-16 model

Typically, a sequence of Convolutional layers followed by MaxPooling layers are used until a 1-dimensional layer is reached.
This is then used as an input to a some connected layers leading to a softmax classification output.


VGG-19
34-layer plain
34-layer residual
image
image
image
output
size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output
size: 112
3x3 conv, 128
3x3 conv, 128
7x7 conv, 64, /2
7x7 conv, 64, /2
pool, /2
pool, /2
pool, /2
output
size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
pool, /2
output
size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128

ResNet

Applications other than images

- Pattern recognition in temporal data
- 1D data
- Any data where there exist recurring patterns correlating to a class

Usefull Links

- https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- https://cs231n.github.io/understanding-cnn/
- https://github.com/keras-team/keras/blob/master/examples/conv_filter_visualization.py
- https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html