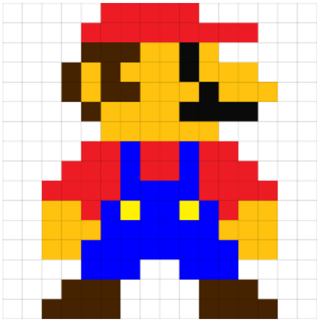
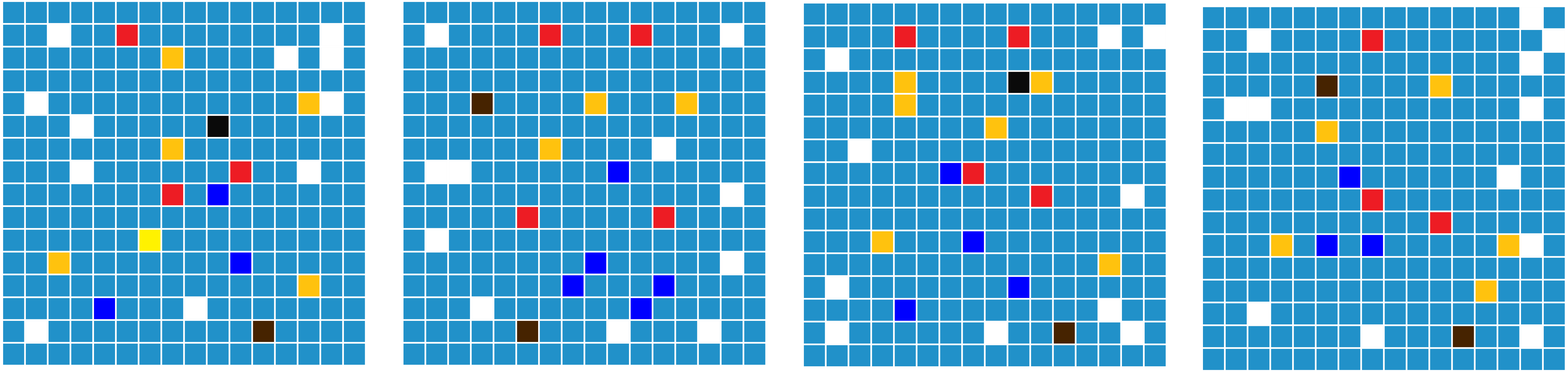
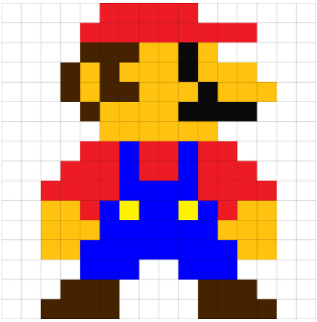
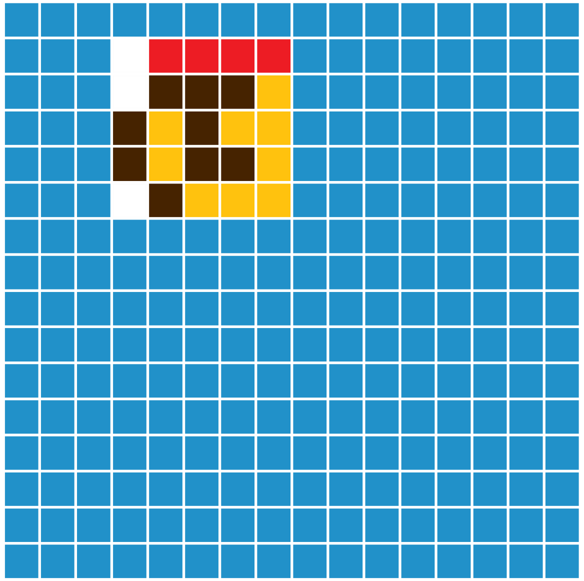
Market Basket Analysis & Lift

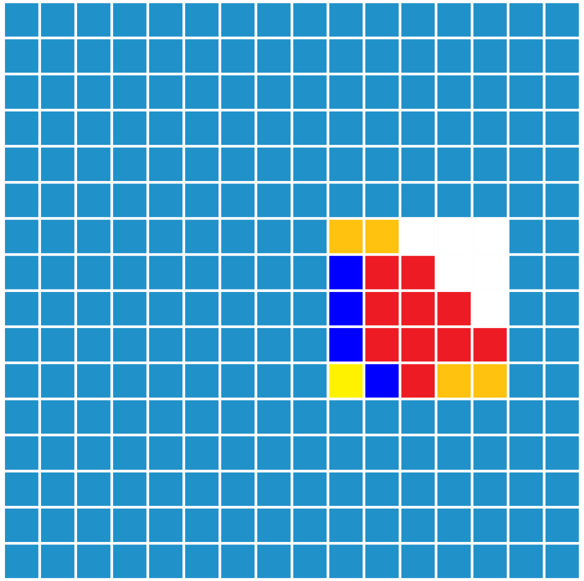


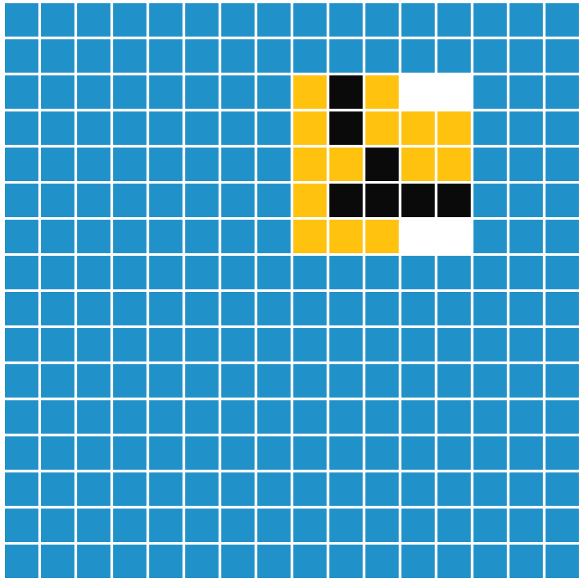


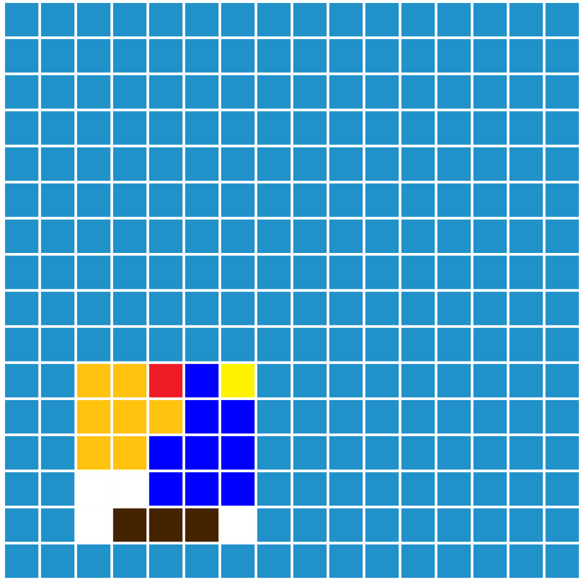
Market Basket Analysis & Lift



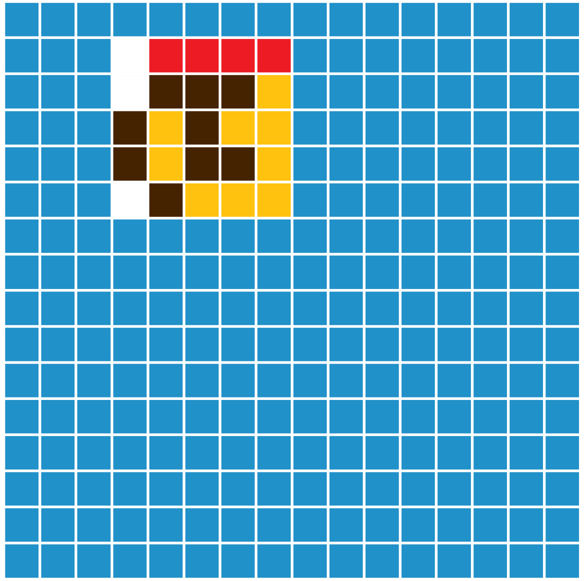


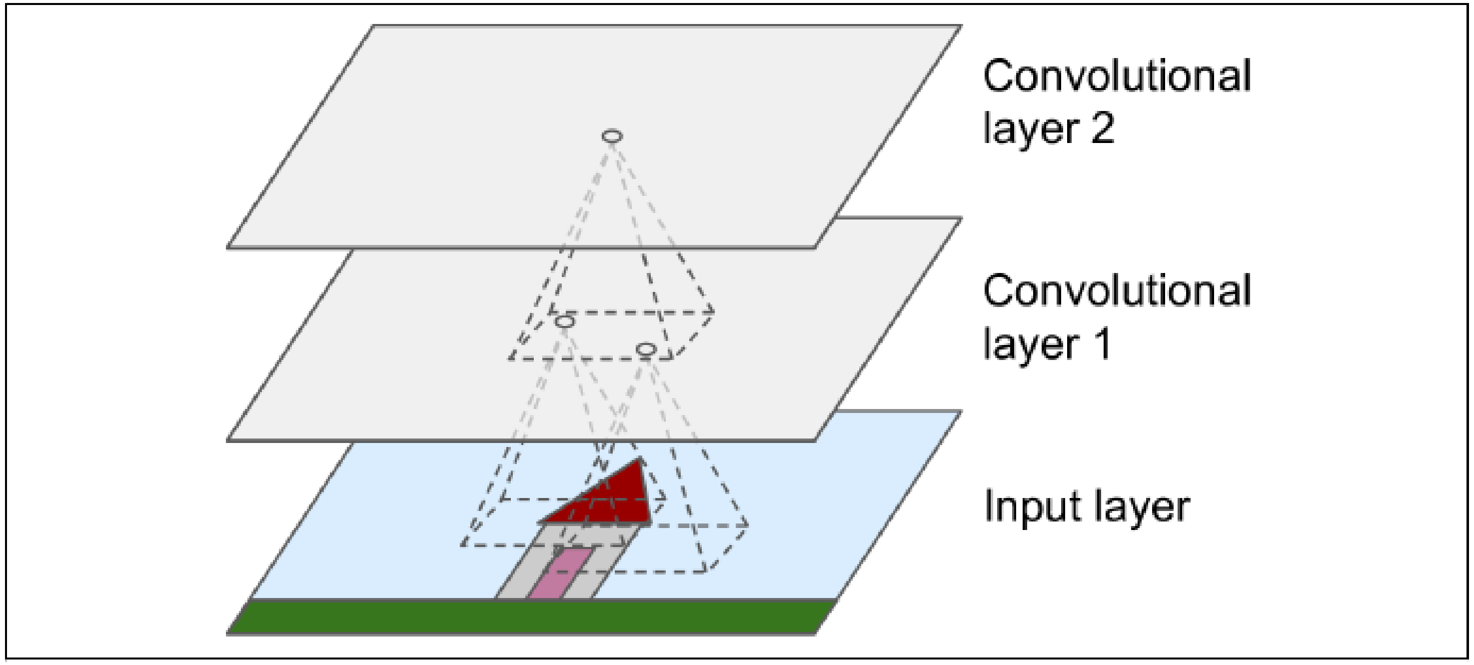




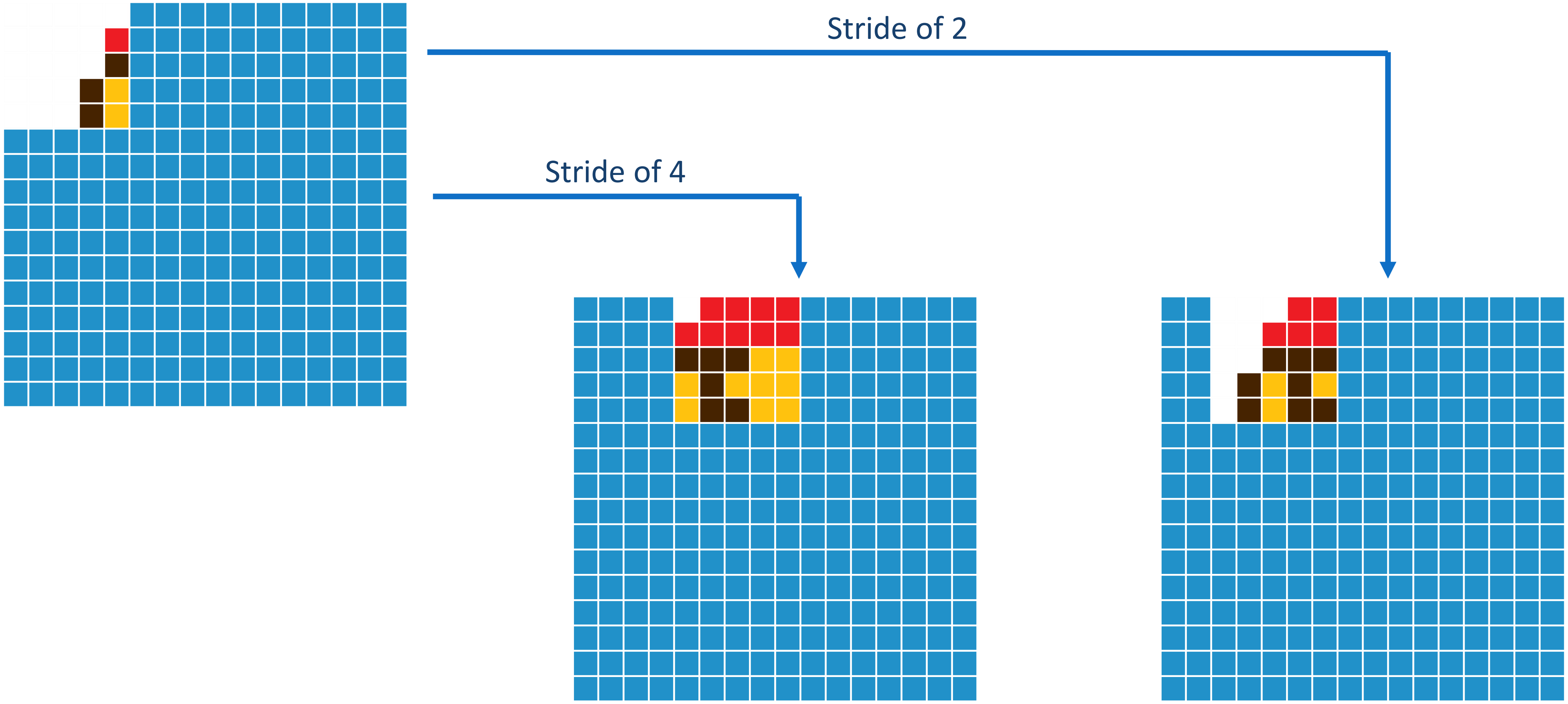


Convolutional Layer

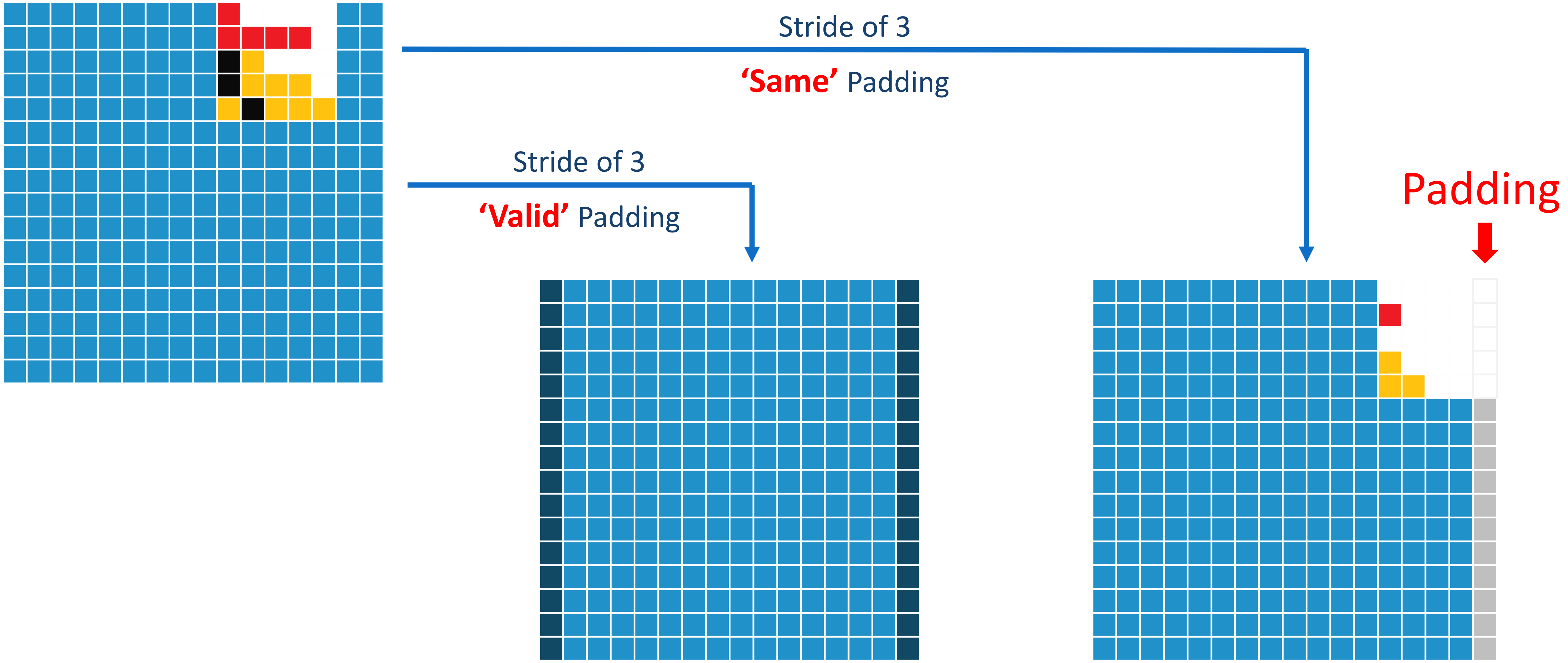




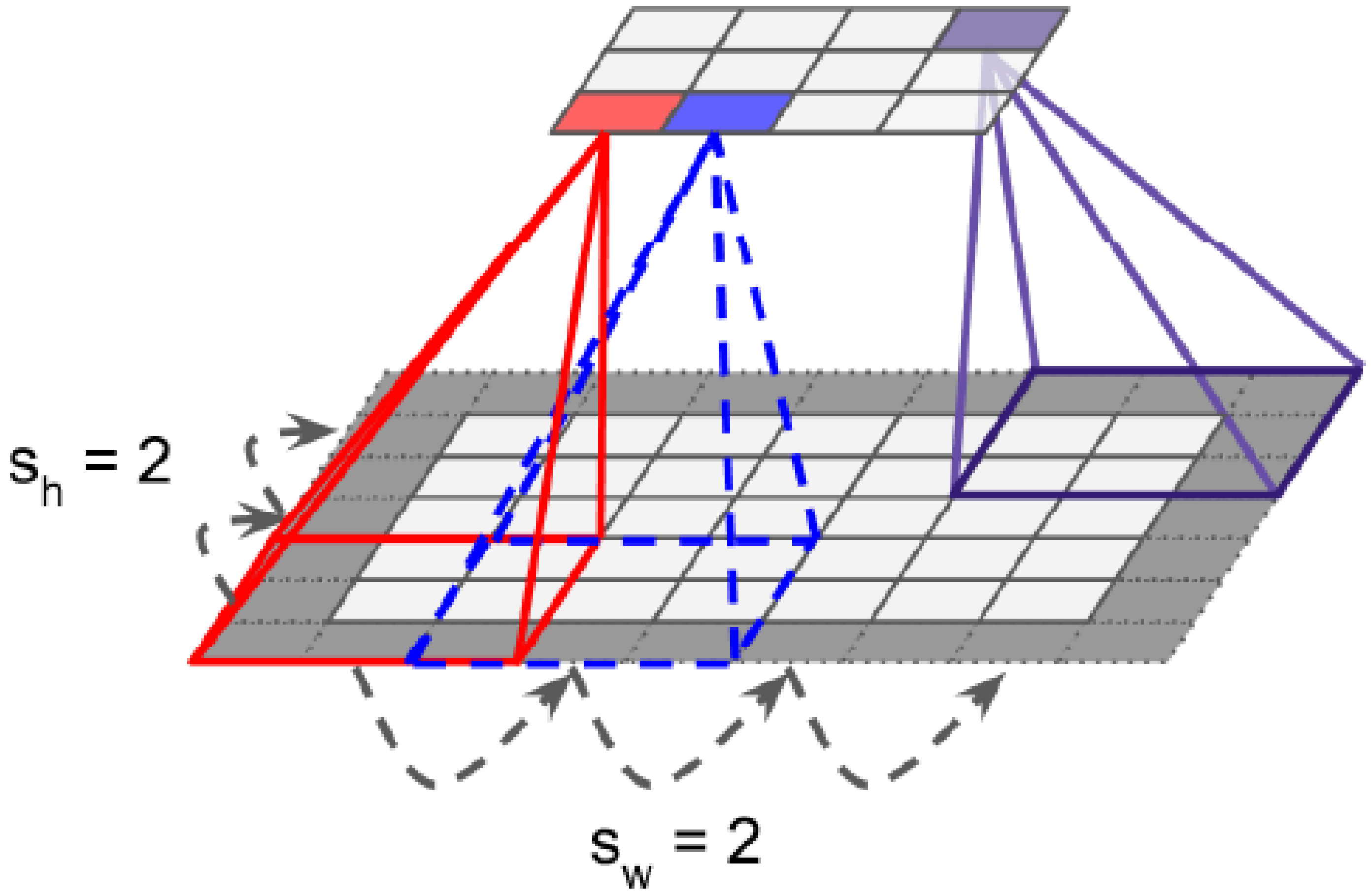
Stride



Padding



Stride / Padding



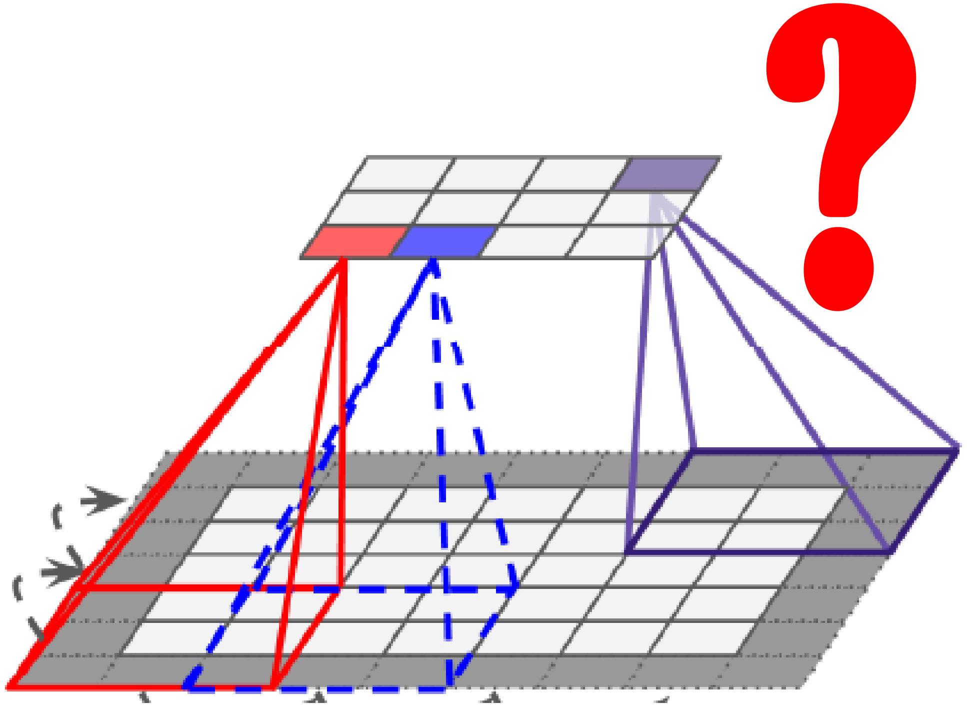
Stride:

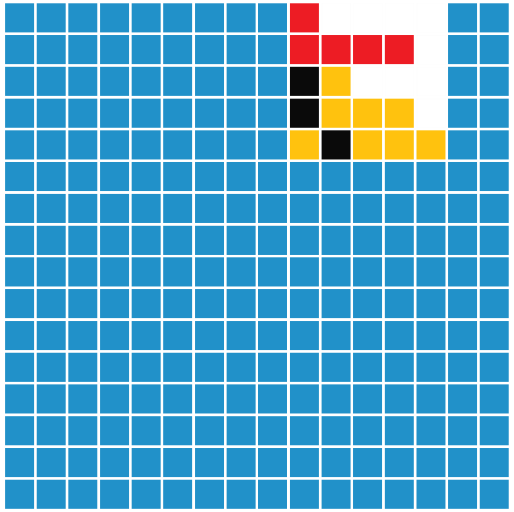
Stride denotes how many steps we are moving in each steps in convolution. By default it is one.

Padding:

Padding is a process of adding zeros to the input matrix symmetrically to maintain the dimension of output as in input

Filter





0.4	0.3	0.1	0.5	0.8
0.1	0.2	0.4	0.1	0.2
0.2	0.3	0.3	0.4	0.4
0.7	0.9	0.8	0.5	0.5
0.9	0.1	0.2	0.1	0.9

Filter

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

✖

1	0	1
0	1	0
1	0	1

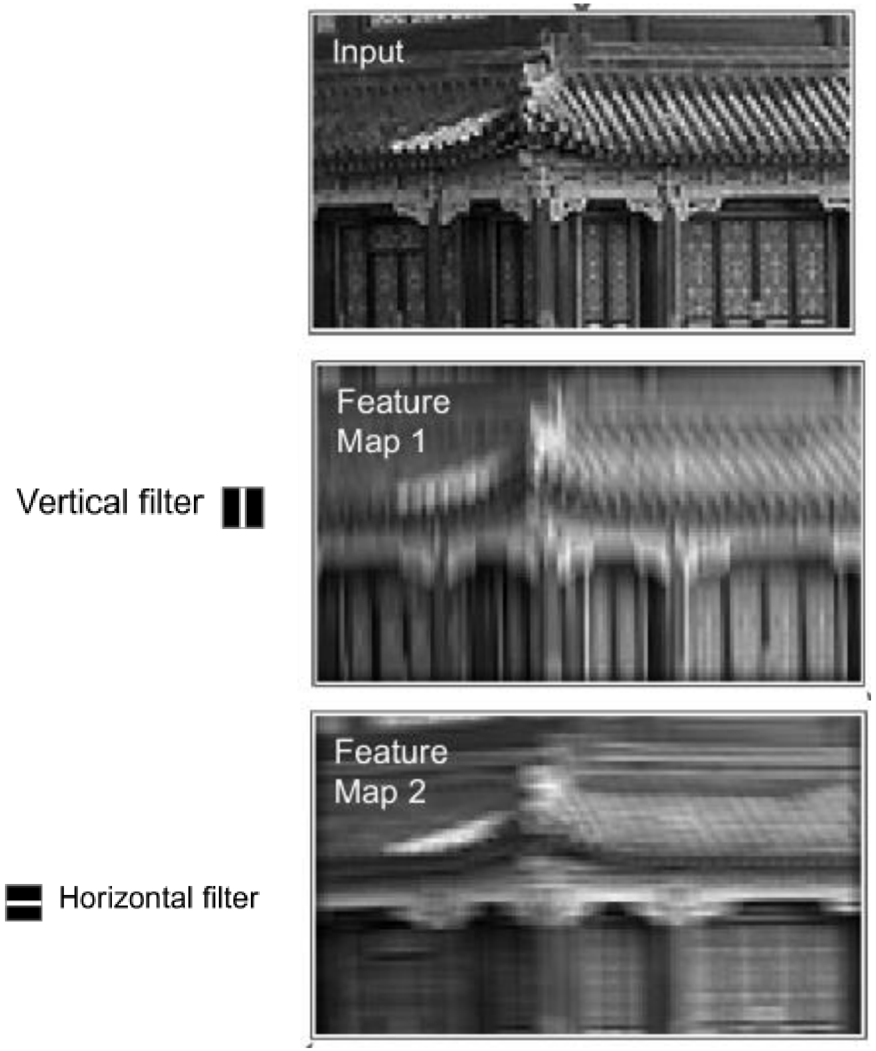
➡

4	3	4
2	4	3
2	3	4

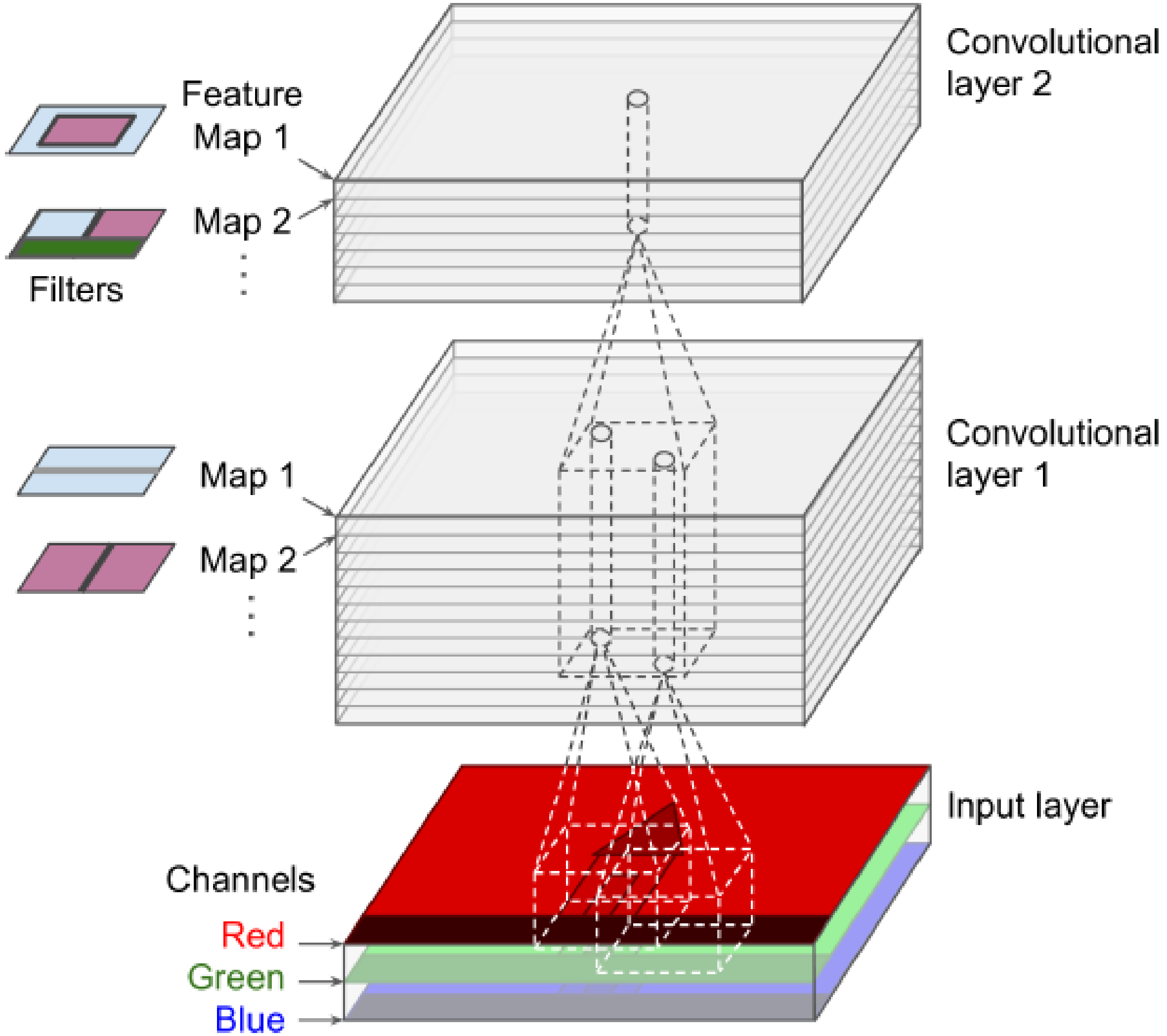
Feature Map

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		



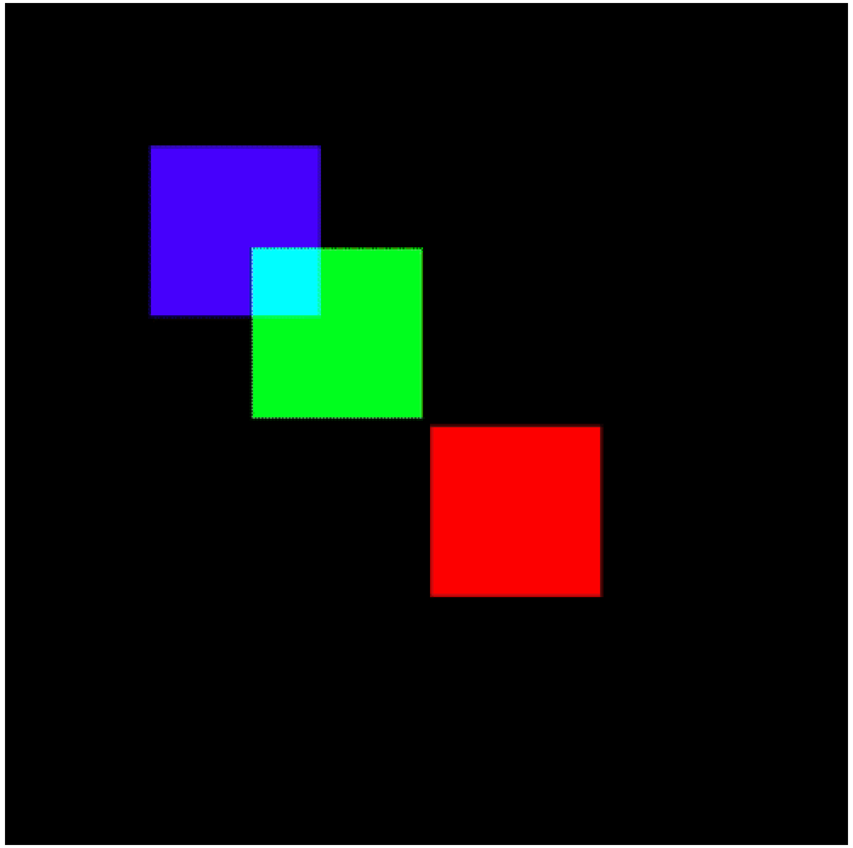
Feature Map



Feature Map:

- The feature map is the output of one filter applied to the previous layer.
- A given filter is drawn across the entire previous layer, moved one pixel at a time.
- Each position results in an activation of the neuron and the output is collected in the feature map
- A Conv Layer can have more than 1 feature map

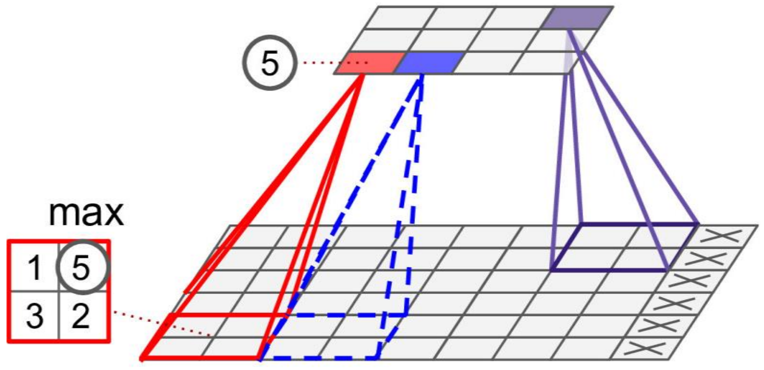
Channels

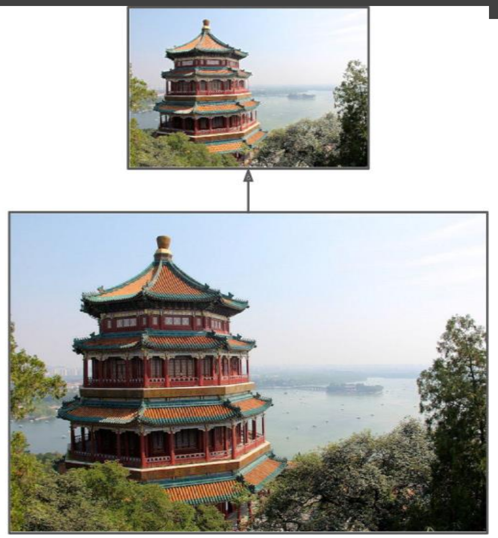


Channels

- Red, green, and blue can be combined in various proportions to obtain any color in the visible spectrum
- Each pixel of any colored image have its own RGB value that is responsible for the color of that pixel
- For example, the following RGB value might be used to create purple:
    - R: 132 (84 in hexadecimal)
    - G: 17 (11 in hexadecimal)
    - B: 170 (AA in hexadecimal)

Pooling Layer





Feature Map

6	6	6	6
4	5	5	4
2	4	4	2
2	4	4	2

Max Pooling

Average Pooling

Sum Pooling

Pooling Layer:

Pooling layers goal is to subsample (i.e., shrink) the input image in order to reduce the computational load, the memory usage, and the number of parameters (thereby limiting the risk of overfitting)

1. Max pooling: The maximum pixel value of the batch is selected.
2. Average pooling: The average value of all the pixels in the batch is selected.

Popular CNN Architectures

WHY

1. To build intuition for good model architecture
2. To use these pre-trained models without retraining for our problem

Popular CNN Architectures

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

The ImageNet Large Scale Visual Recognition Challenge or ILSVRC for short is an annual competition helped between 2010 and 2017 in which challenge tasks use subsets of the ImageNet dataset.

The goal of the challenge was to both

1. Promote the development of better computer vision techniques
2. To benchmark the state of the art

Popular CNN Architectures

Popular ILSVRC submissions

Year	CNN	Developed By	Error rates	No. of parameters
1998	LeNet	Yann LeCun et al		60 thousand
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	15.3%	60 million
2013	ZFNet	Matthew Zeiler, Rob Fergus	14.8%	
2014	GoogLeNet	Google	6.67%	4 million
2014	VGGNet	Simonyan, Zisserman	7.3%	138 million
2015	ResNet	Kaiming He	3.6%	

LeNet-5

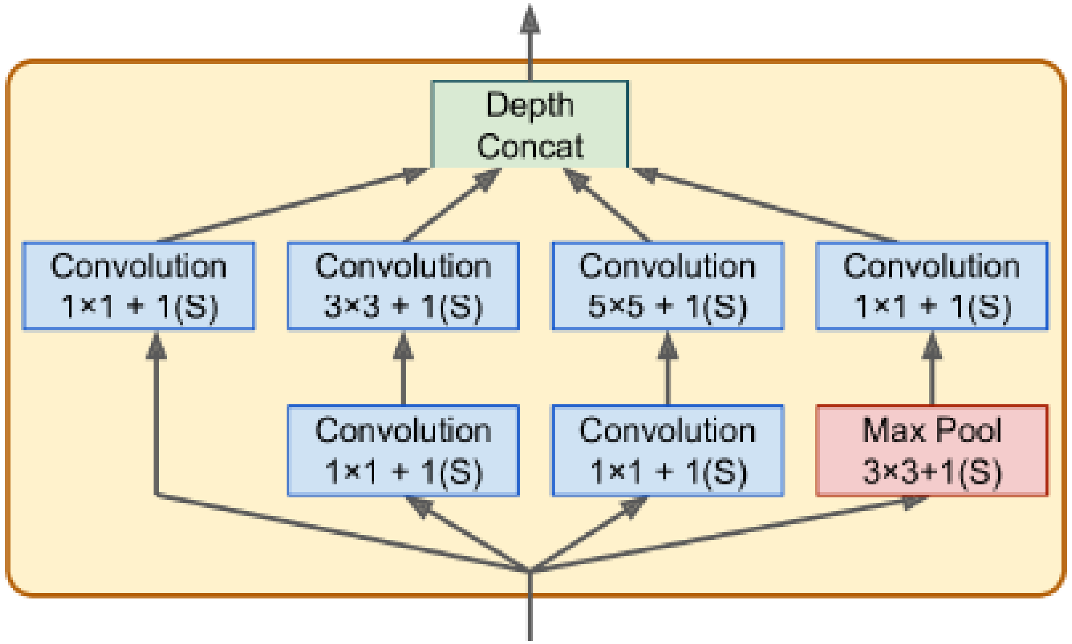
Most popular CNN architecture
Created by LeCunn in 1998

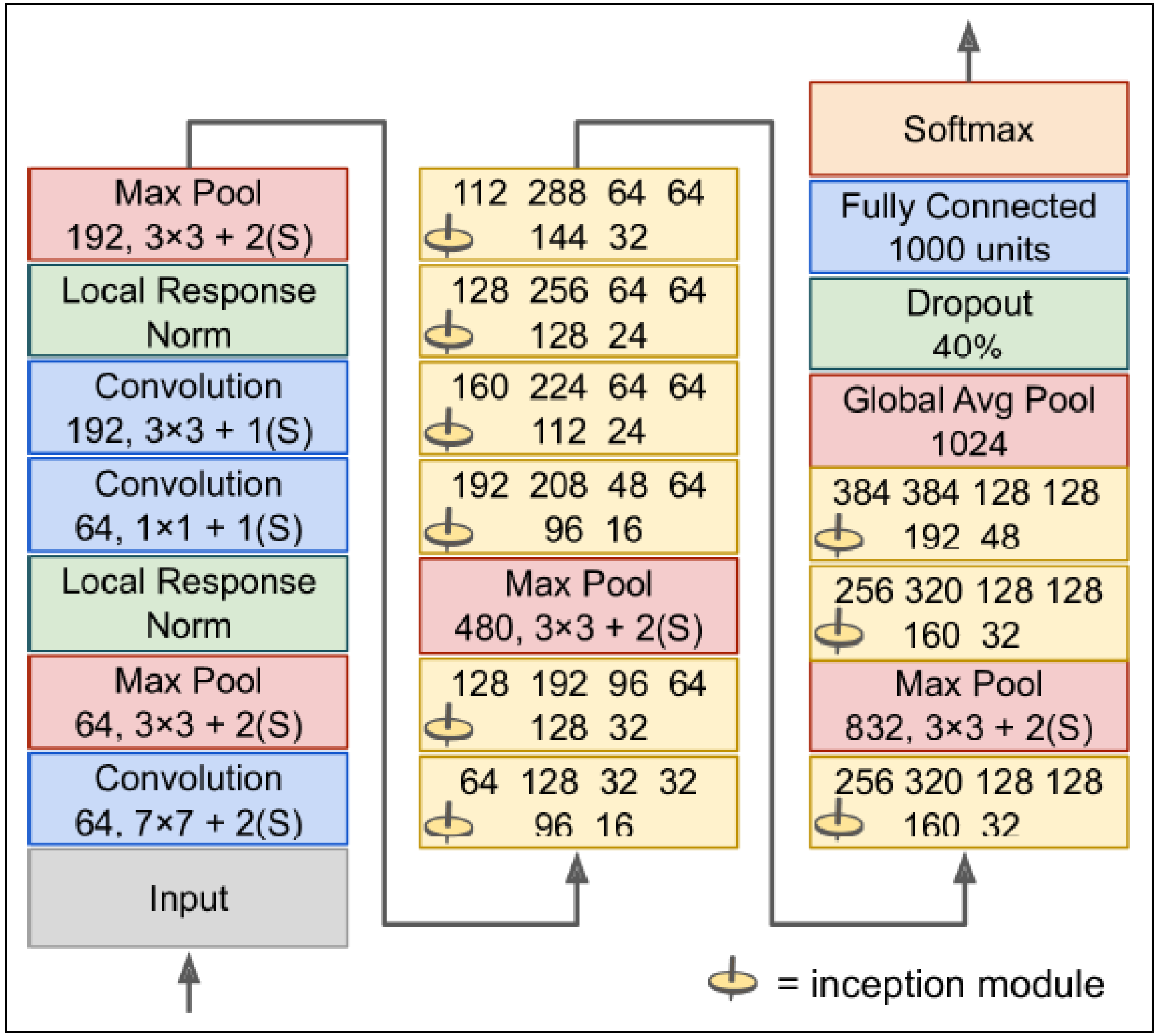
Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

VGG16

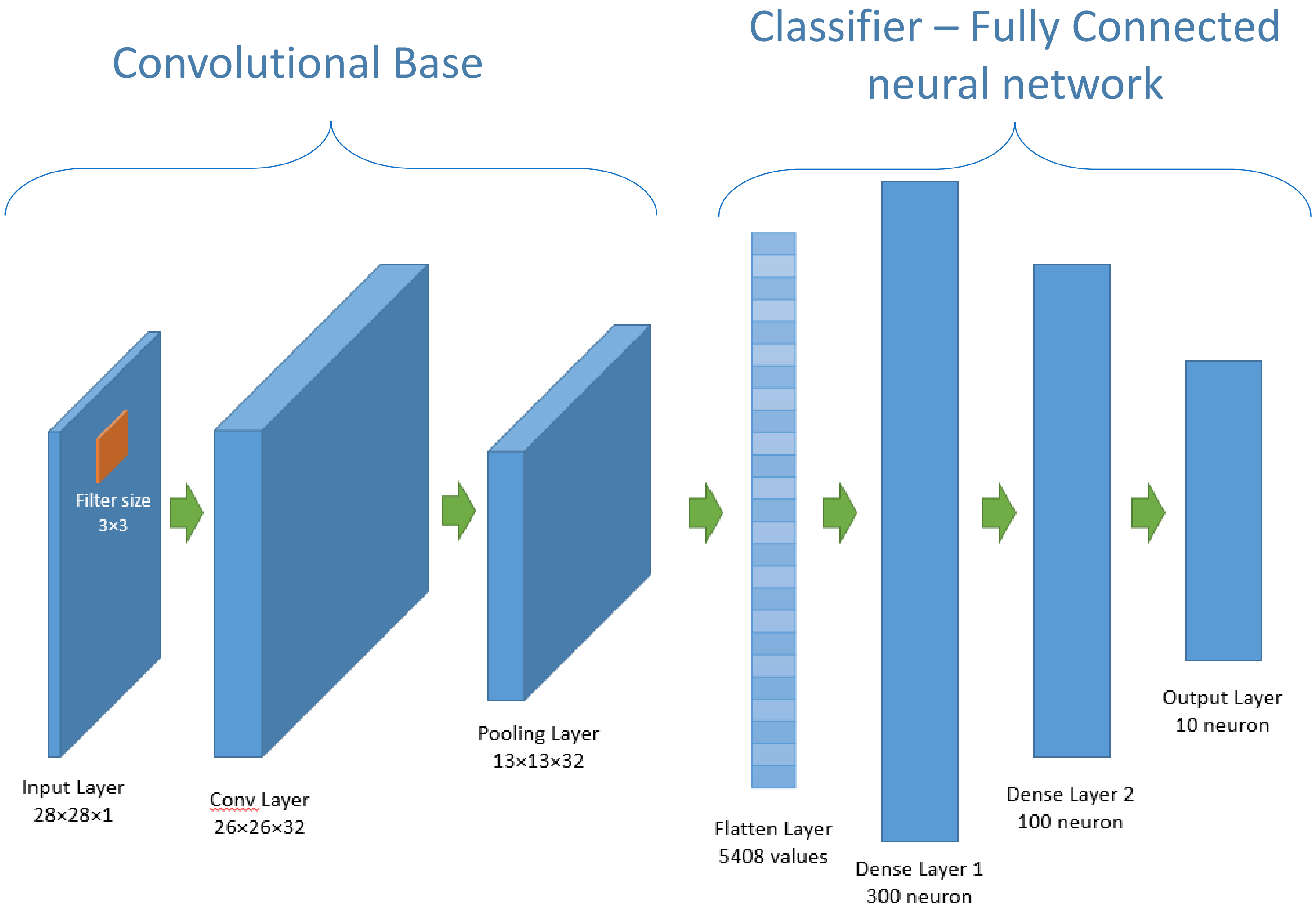
Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

GoogLeNet


Depth Concat
Convolution 1×1 + 1(S)
Convolution 3×3 + 1(S)
Convolution 5×5 + 1(S)
Convolution 1×1 + 1(S)
Convolution 1×1 + 1(S)
Convolution 1×1 + 1(S)
Max Pool 3×3+1(S)


Input
Convolution 64, 7×7 + 2(S)
Max Pool 64, 3×3 + 2(S)
Local Response Norm
Convolution 64, 1×1 + 1(S)
Convolution 192, 3×3 + 1(S)
Local Response Norm
Max Pool 192, 3×3 + 2(S)
64 128 32 32 96 16
128 192 96 64 128 32
Max Pool 480, 3×3 + 2(S)
192 208 48 64 96 16
160 224 64 64 112 24
128 256 64 64 128 24
112 288 64 64 144 32
256 320 128 128 160 32
Max Pool 832, 3×3 + 2(S)
256 320 128 128 160 32
384 384 128 128 192 48
Global Avg Pool 1024
Dropout 40%
Fully Connected 1000 units
Softmax
= inception module

Transfer Learning



Convolutional base can be re-used with new classifier

Advantages

- Saves a lot of training time
- Proven models with good accuracy
- Models trained on large datasets – better features extracted
- Easy to use