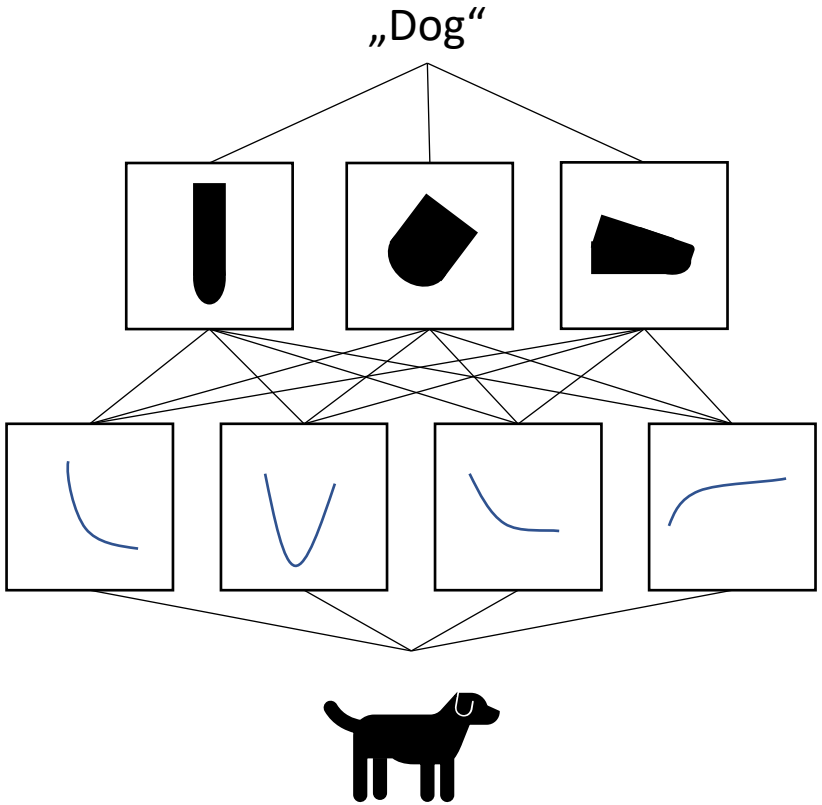
Convolutional Neural Networks 101

Bert Gollnick

Convolutional Neural Networks

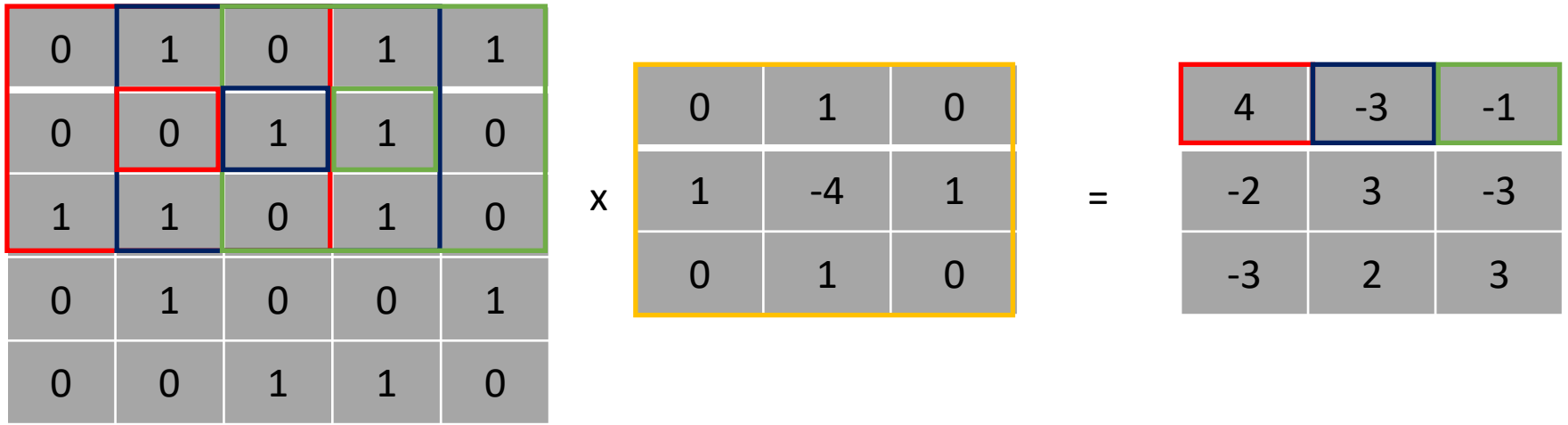
Introduction

- Type of deep learning network
- Most commonly applied for computer vision
- CNNs learn local patterns
- Local patterns translational invariant
- Layers learn different hierarchical patterns (from simple to complex)



Convolutional Neural Networks

Convolution



Input Image Matrix

Convolutional Filter
(Edge Detector)

Feature Map

Convolutional Neural Networks

Convolution - Example

x

0	1	0
1	-4	1
0	1	0

=

Input Image

Edge Detector

Feature Map

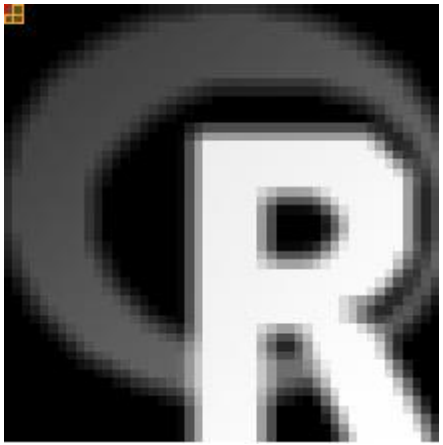
Convolutional Neural Networks

Convolution - Example

x

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

=



Input Image

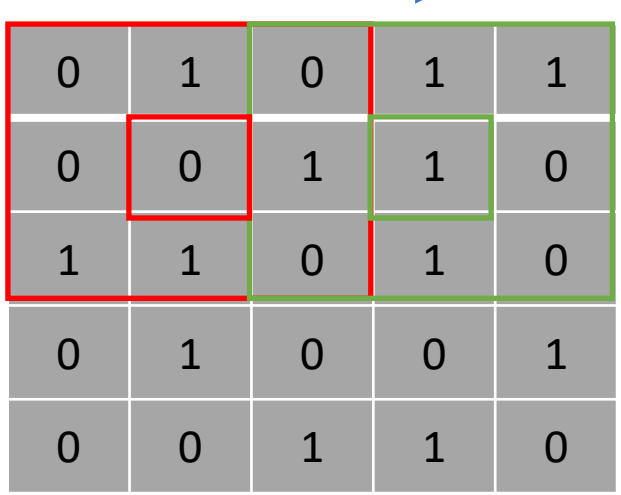
Blur Filter

Feature Map

Convolutional Neural Networks

Convolution - Stride

- Stride equals step size
- Example: Stride = 2



Input Image Matrix

x

0	1	0
1	-4	1
0	1	0

Convolutional Filter
(Edge Detector)

=

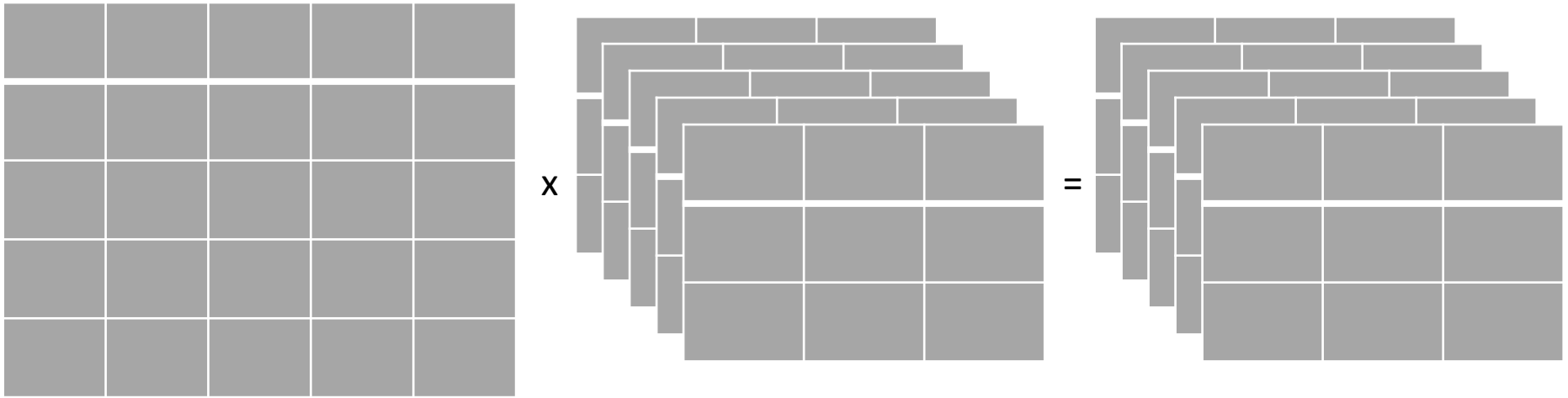
4	-1
-3	3

Feature Map

Convolutional Neural Networks

Convolutional Layer

Feature Maps



Input Image Matrix

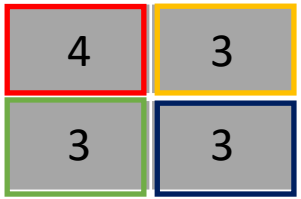
Convolutional Filters

Convolutional Layer

Convolutional Neural Networks

Max Pooling

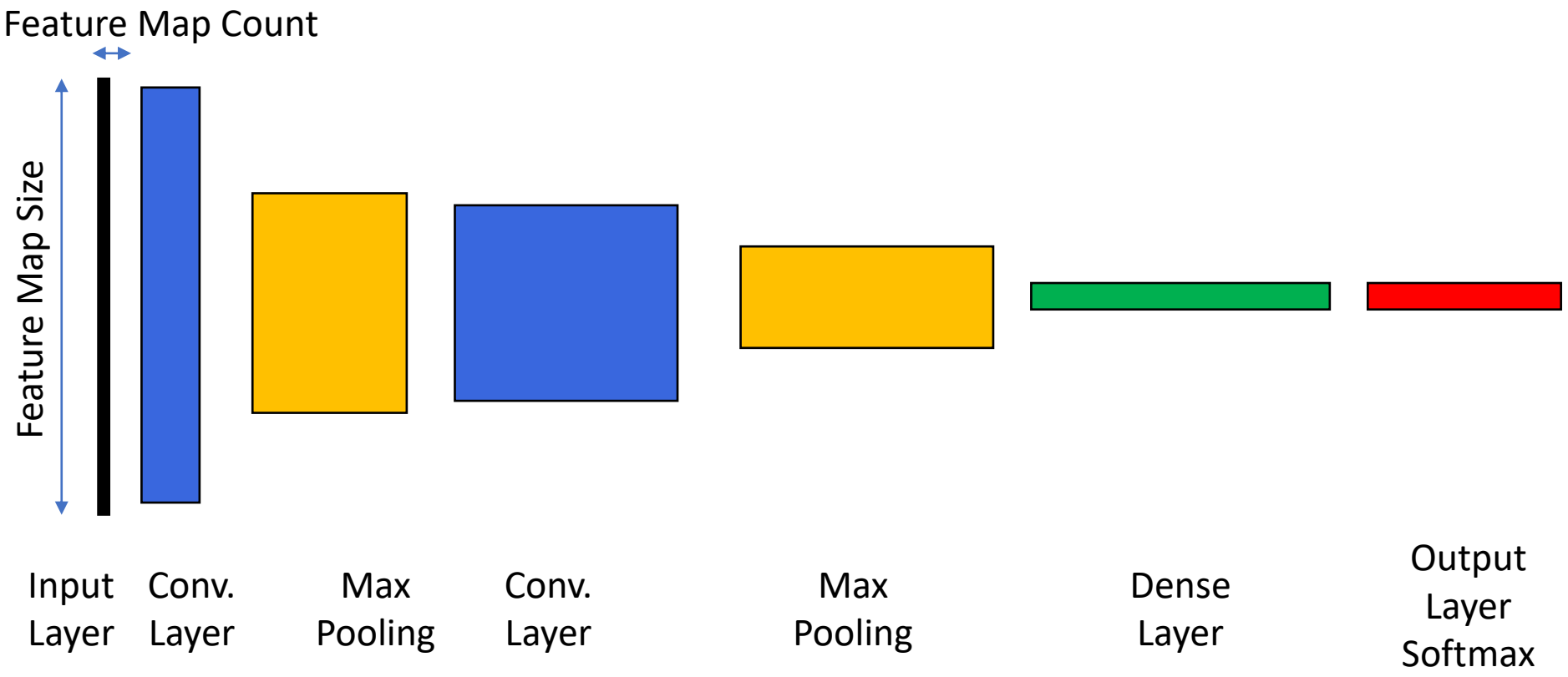
Feature Map



Max Pooling

Convolutional Neural Networks

Example Network Setup



Convolutional Neural Networks

Example Network Setup

- Count of feature maps increase with layers from (from 32 to 64)
- Size of feature maps decreases (from 148x148 to 16x16)
- Very typical for CNNs

Layer (type)	Output Shape	Param #
conv2d_19 (Conv2D)	(None, 148, 148, 32)	320
max_pooling2d_16 (MaxPooling2	(None, 74, 74, 32)	0
conv2d_20 (Conv2D)	(None, 72, 72, 32)	9248
max_pooling2d_17 (MaxPooling2	(None, 36, 36, 32)	0
conv2d_21 (Conv2D)	(None, 34, 34, 64)	18496
conv2d_22 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d_18 (MaxPooling2	(None, 16, 16, 64)	0
dropout_17 (Dropout)	(None, 16, 16, 64)	0
flatten_7 (Flatten)	(None, 16384)	0
dense_13 (Dense)	(None, 512)	8389120
dropout_18 (Dropout)	(None, 512)	0
dense_14 (Dense)	(None, 10)	5130

Total params: 8,459,242
Trainable params: 8,459,242
Non-trainable params: 0

Convolutional Neural Networks

Advantages / Disadvantages

\+

- Most powerful technique for computer vision tasks
- Achieves high quality predictions

\-

- Many parameters
- Requires a lot of experience
- Very computationally expensive