

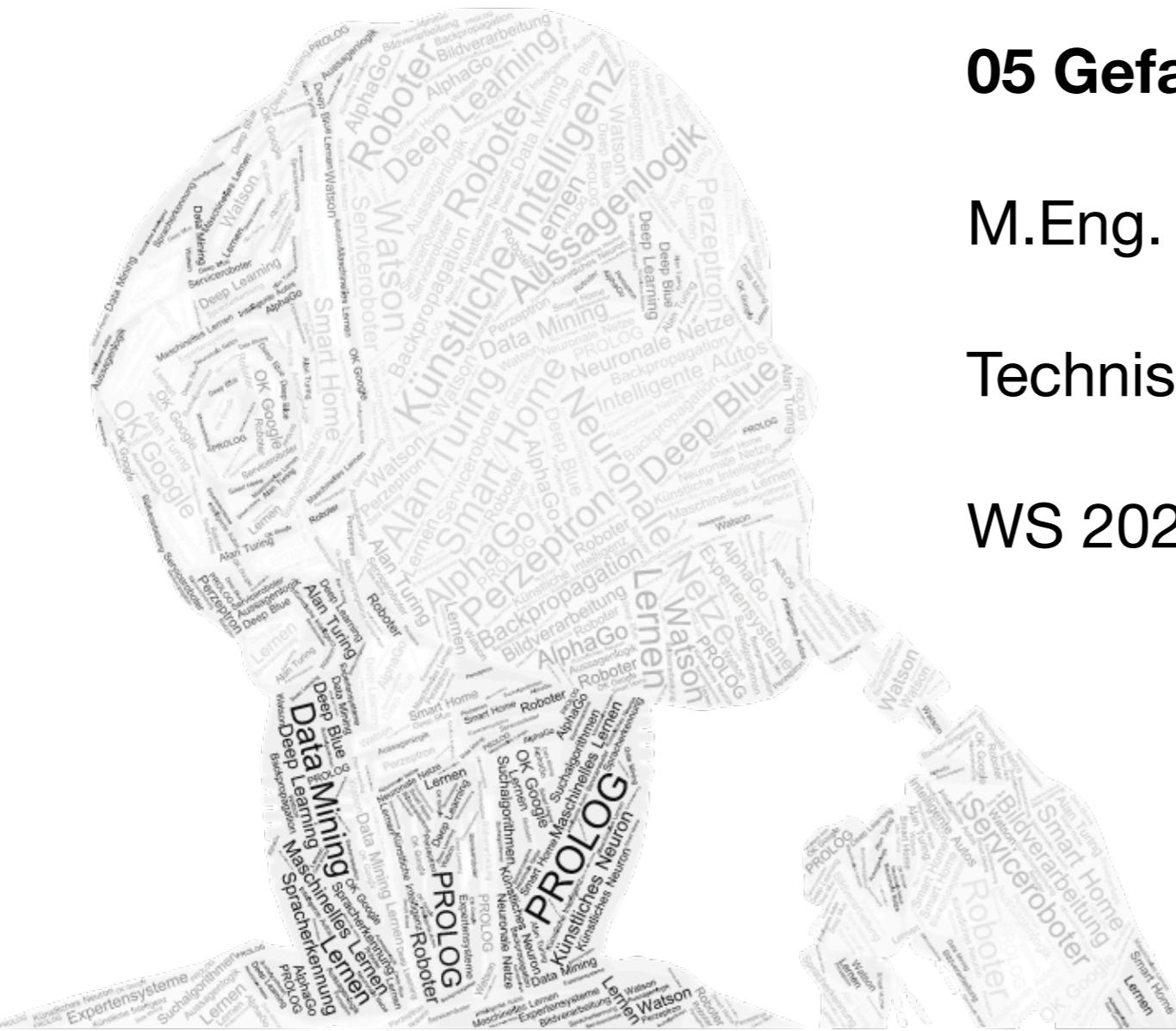
Künstliche Intelligenz

05 Gefaltete Neuronale Netzwerke

M.Eng. Janine Breßler

Technische Hochschule Wildau

WS 2025/2026



Gefaltete Neuronale Netze

Motivation

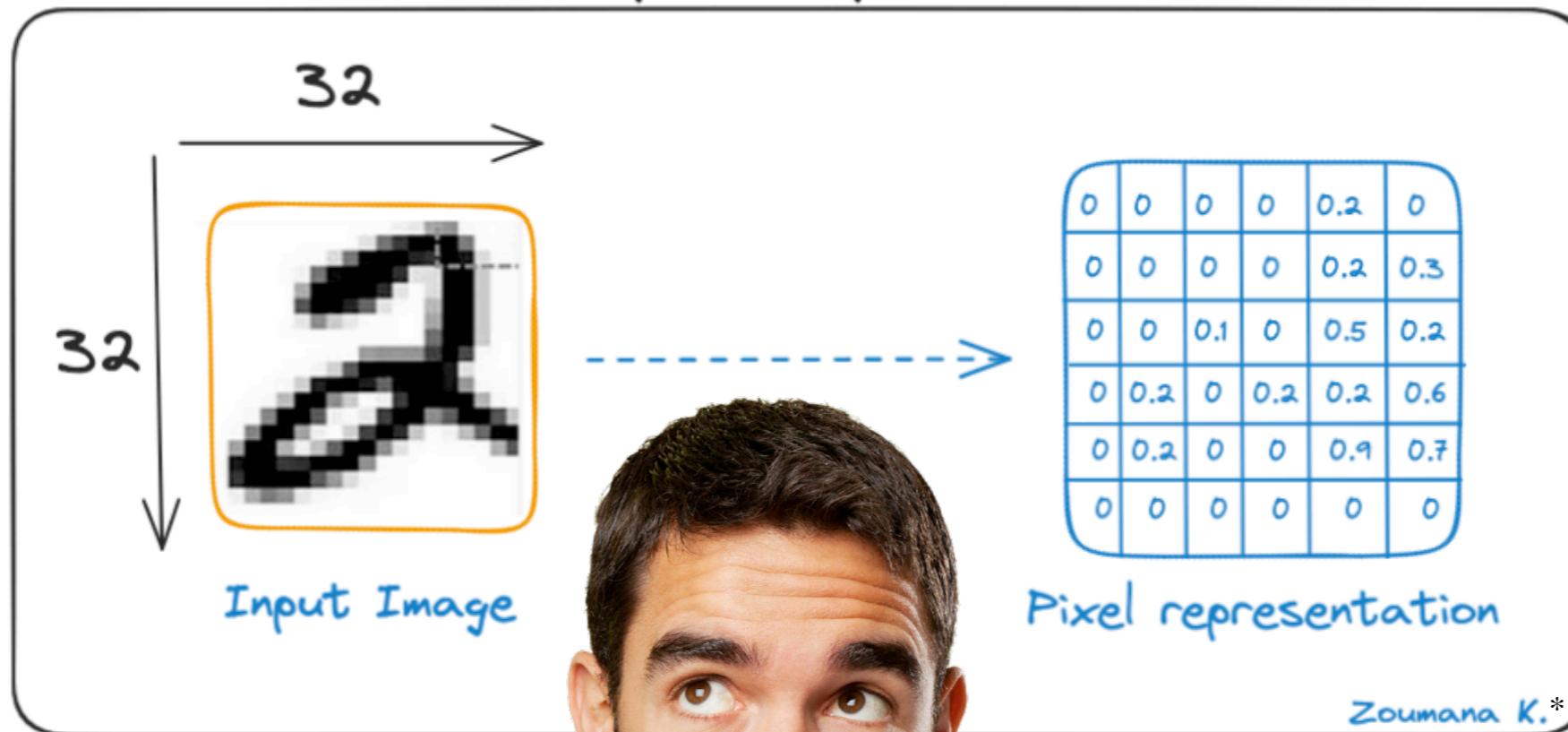
Gefaltete Neuronale Netze, auch Convolutional Neural Networks (CNN), haben einen speziellen Aufbau, d.h. es gibt unterschiedliche Arten von Schichten mit unterschiedlichen Aufgaben.

CNNs oder auch ConvNets erzielen besonders in den folgenden Disziplinen gute Ergebnisse:

- Bildverarbeitung
- Verarbeitung von Sprache
- Verarbeitung von Audiosignalen

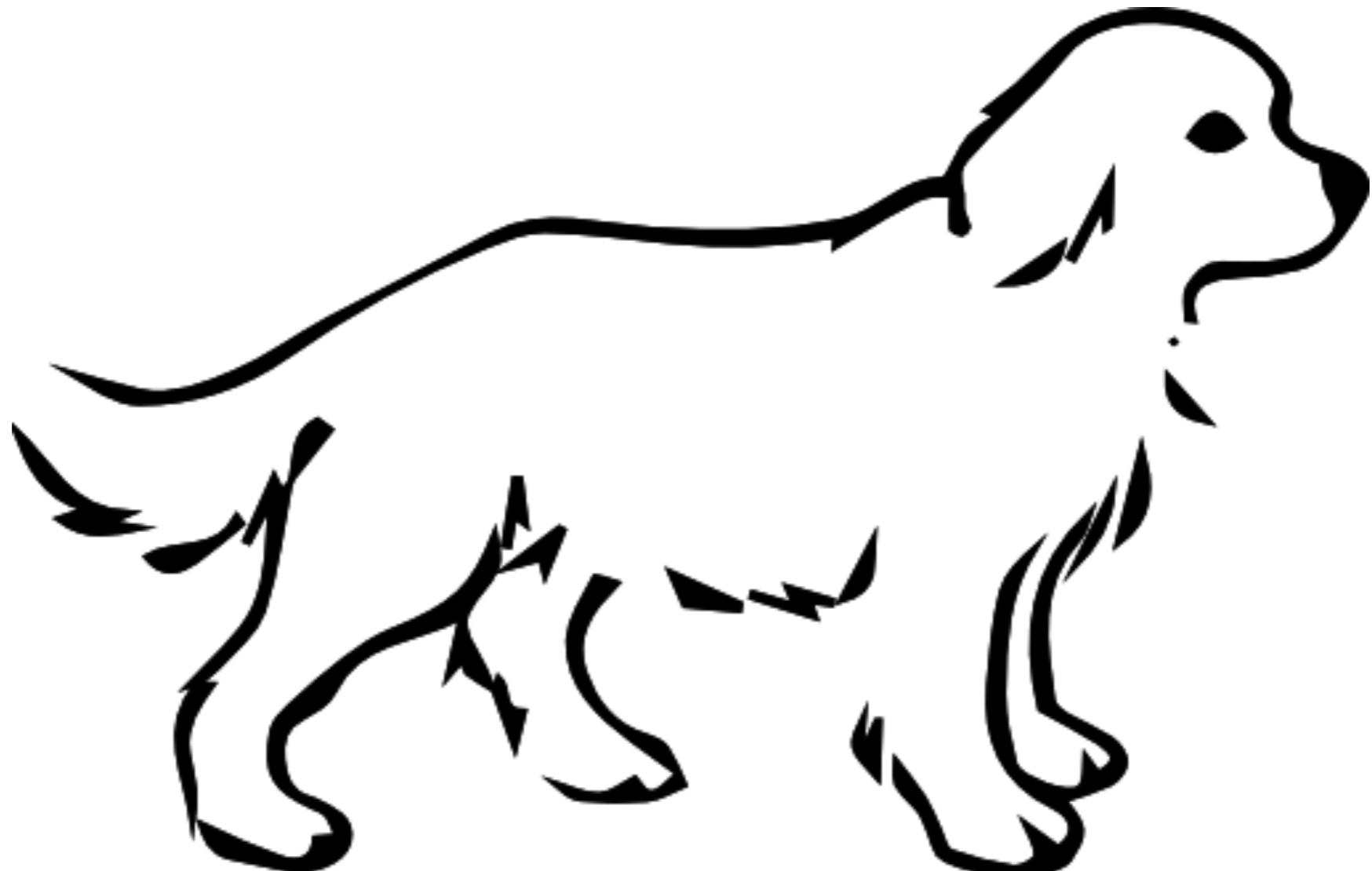
Warum Convolutional Neural Networks?

Illustration of the input image and its pixel representation



* Zoumana Keita, Data Scientist

Warum Convolutional Neural Networks?

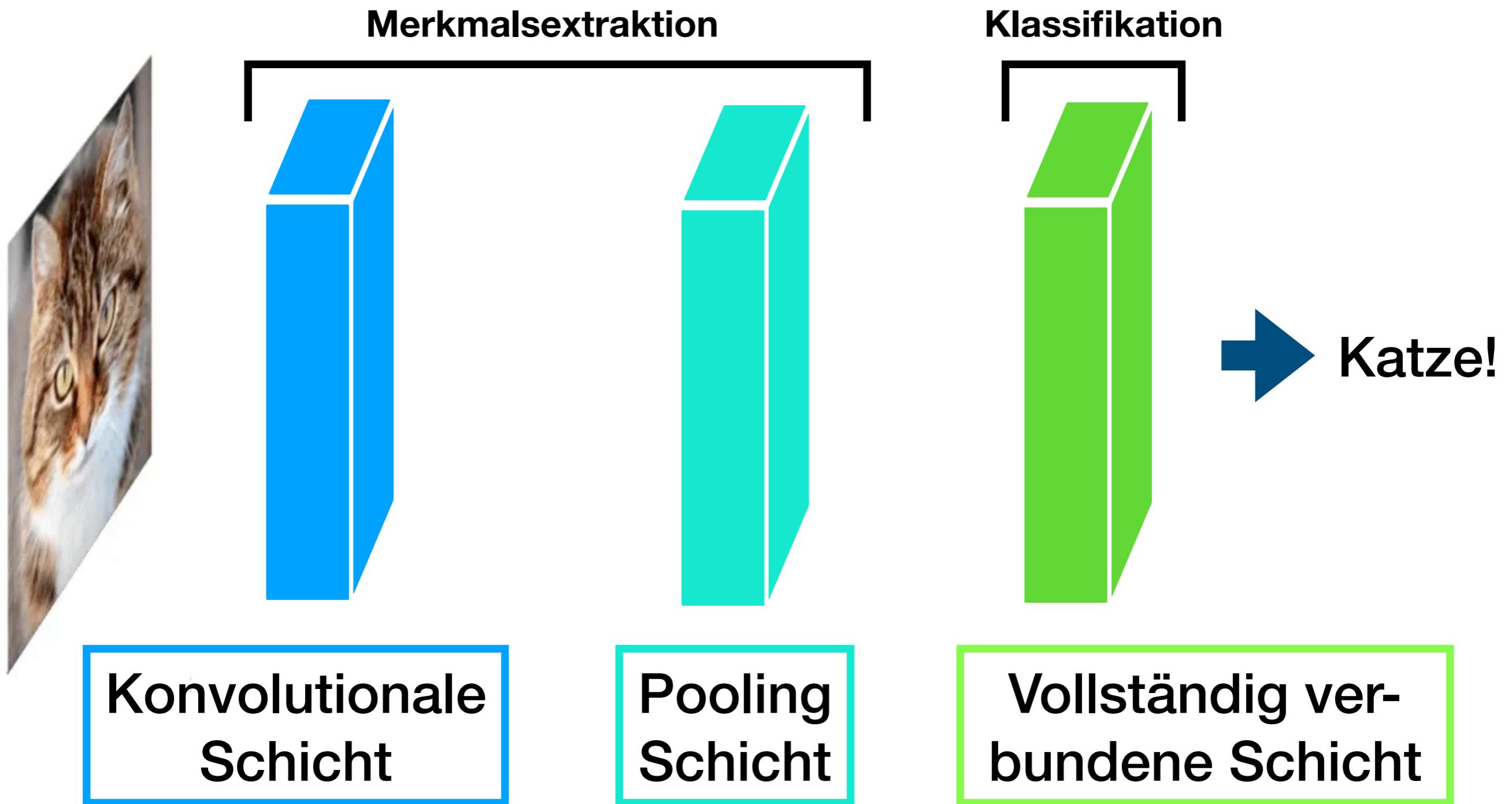


[Digital Ocean](#)

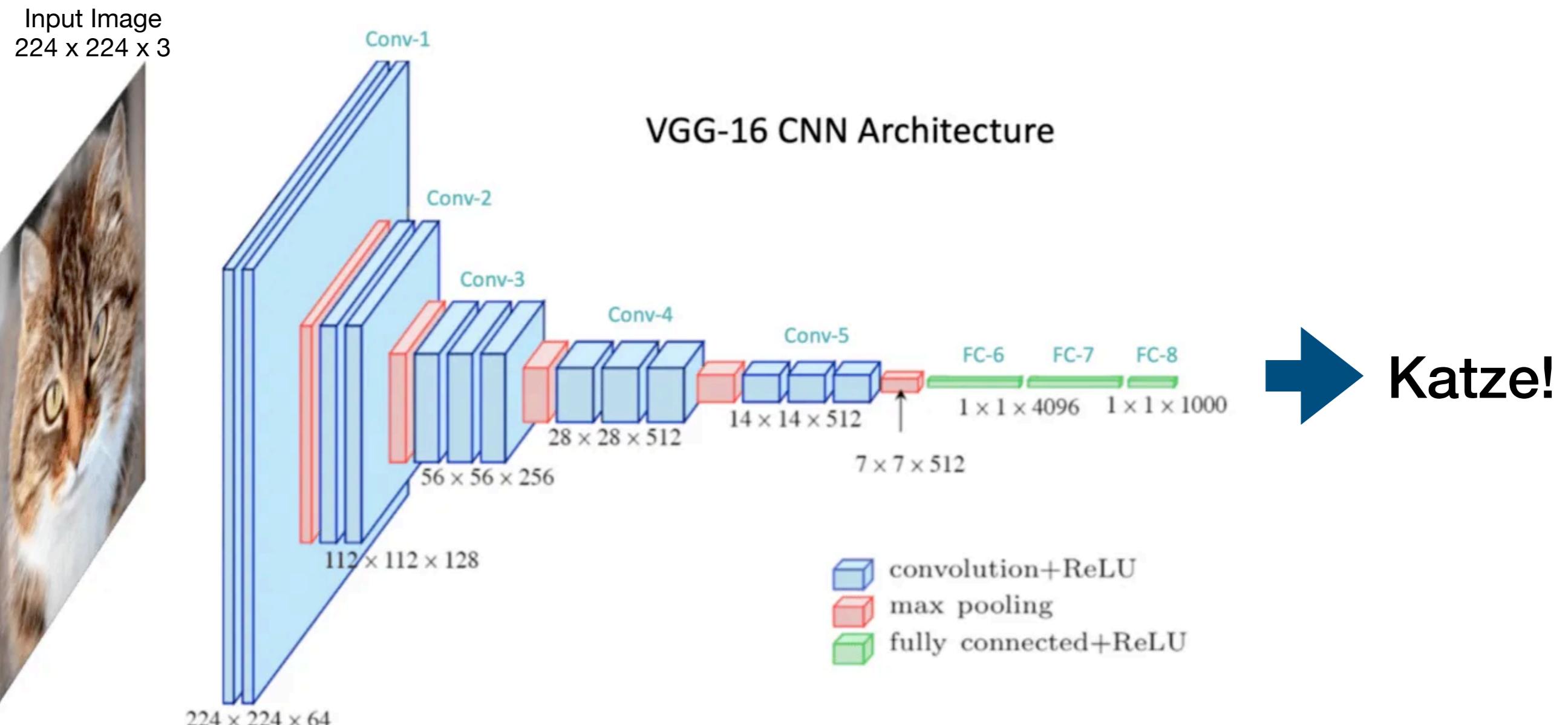
Warum Convolutional Neural Networks?



Komponenten von CNNs

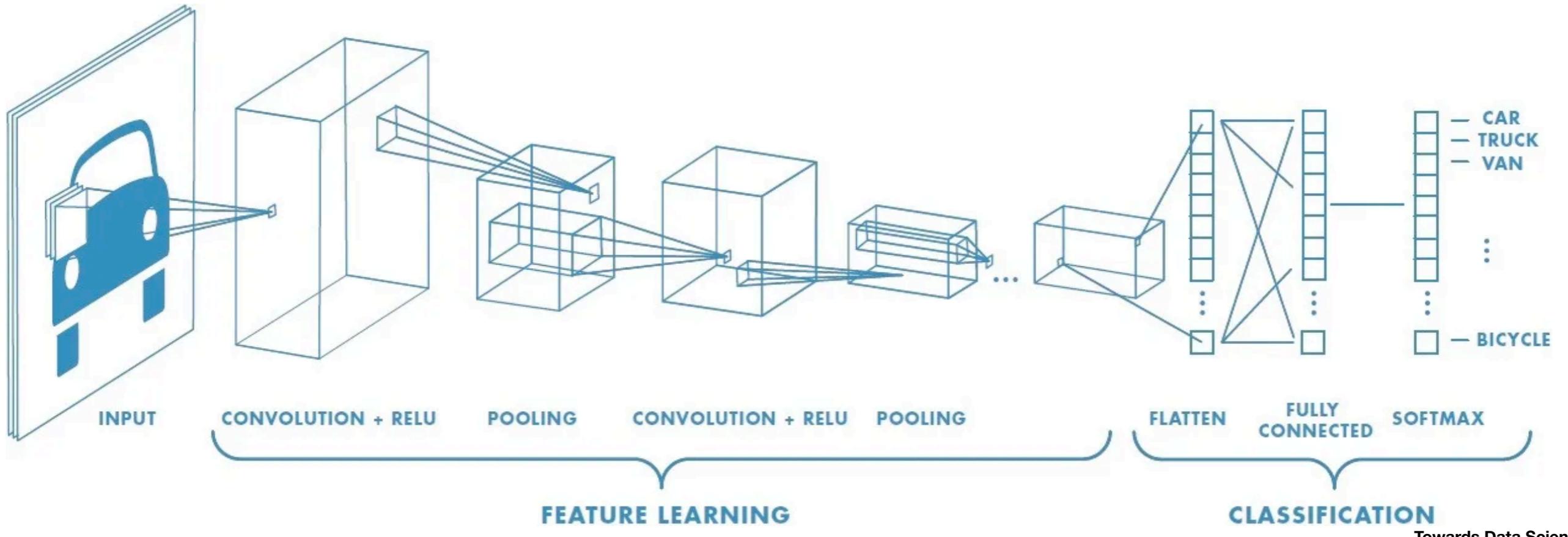


Komponenten von CNNs

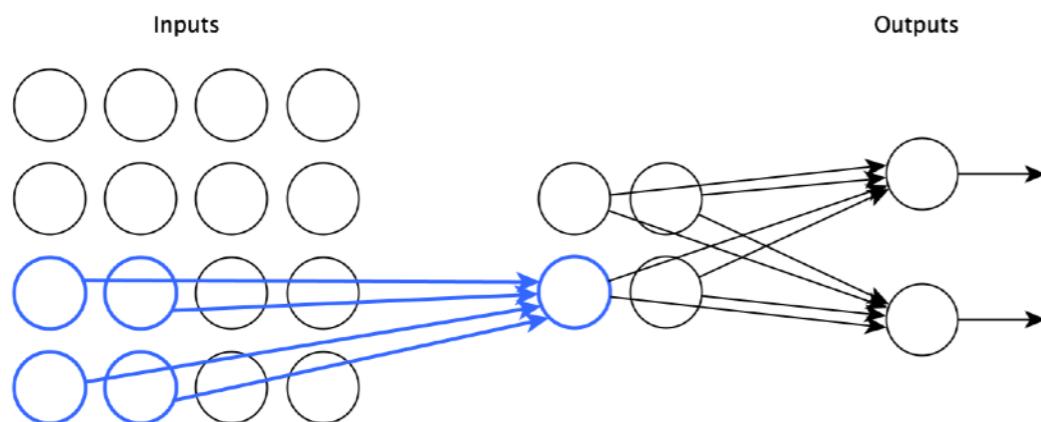


Learn Open CV

Komponenten von CNNs

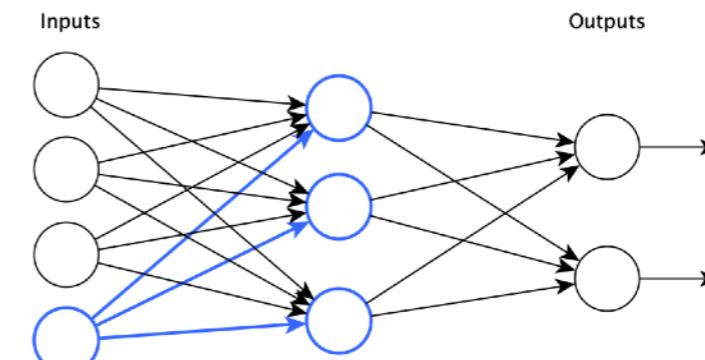


Gefaltetes Neuronales Netzwerk



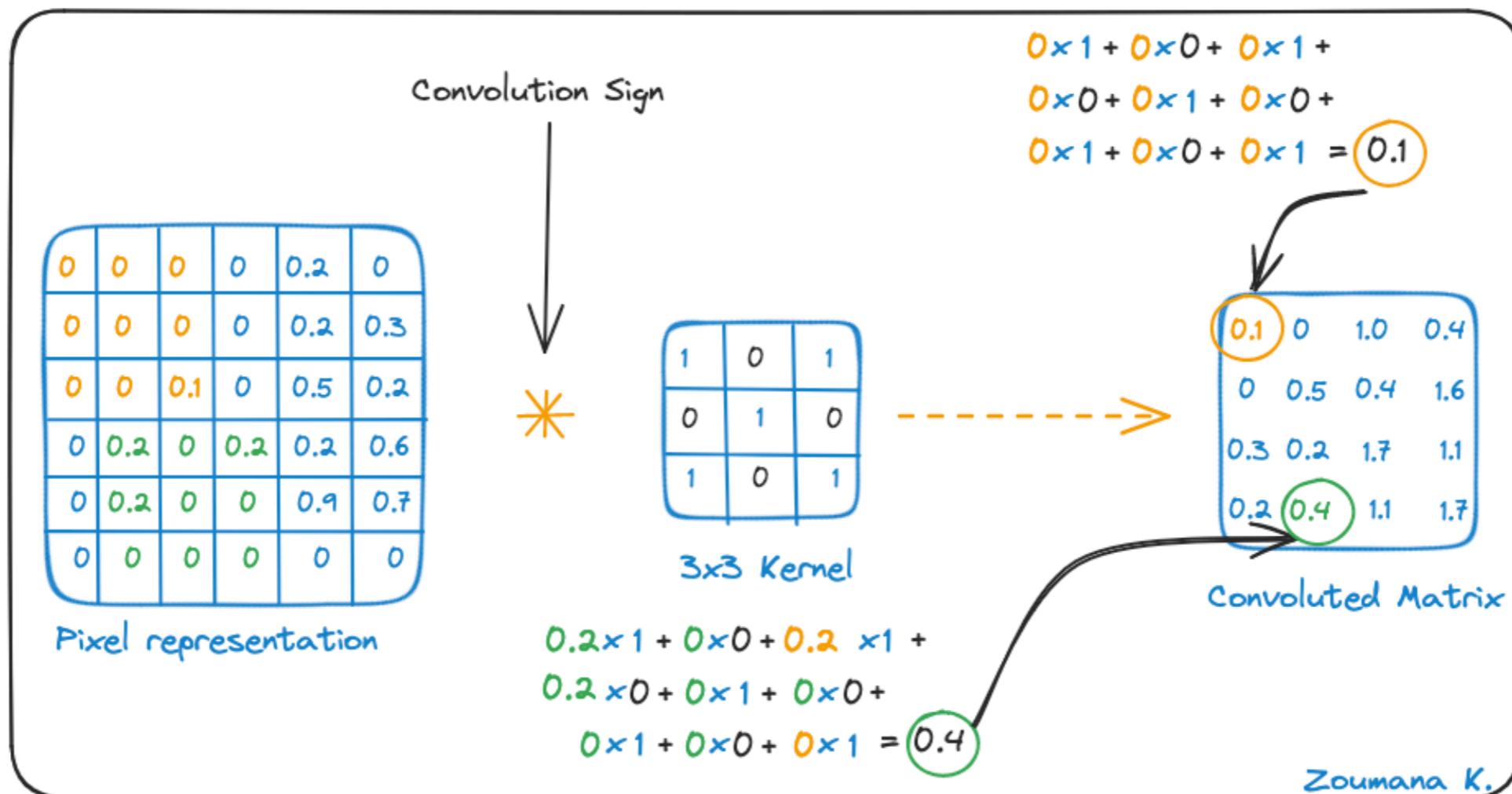
Reduktion der Parameter!

Voll vermaschtes Neuronales Netzwerk



Konvolutionale Schicht

Application of the convolution task using
a stride of 1 with 3x3 kernel



Filter / Kernel

Die Filter haben die Aufgabe, Kanten, Formen oder Texturen zu erkennen:

Beispiel Kantenerkennung mit Hilfe des Prewitt-Filters:

Prewitt Filters

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

horizontal

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

vertical

[Digital Ocean](#)

Filter / Kernel

Beispiel Kantenerkennung mit Hilfe des Prewitt-Filters:

Prewitt Filters

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

horizontal

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

vertical

[Digital Ocean](#)

original



convolved



horizontaler Prewitt-Filter

[Digital Ocean](#)

Filter / Kernel

Beispiel Kantenerkennung mit Hilfe des Prewitt-Filters:

Prewitt Filters

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

horizontal

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

vertical

original



convolved

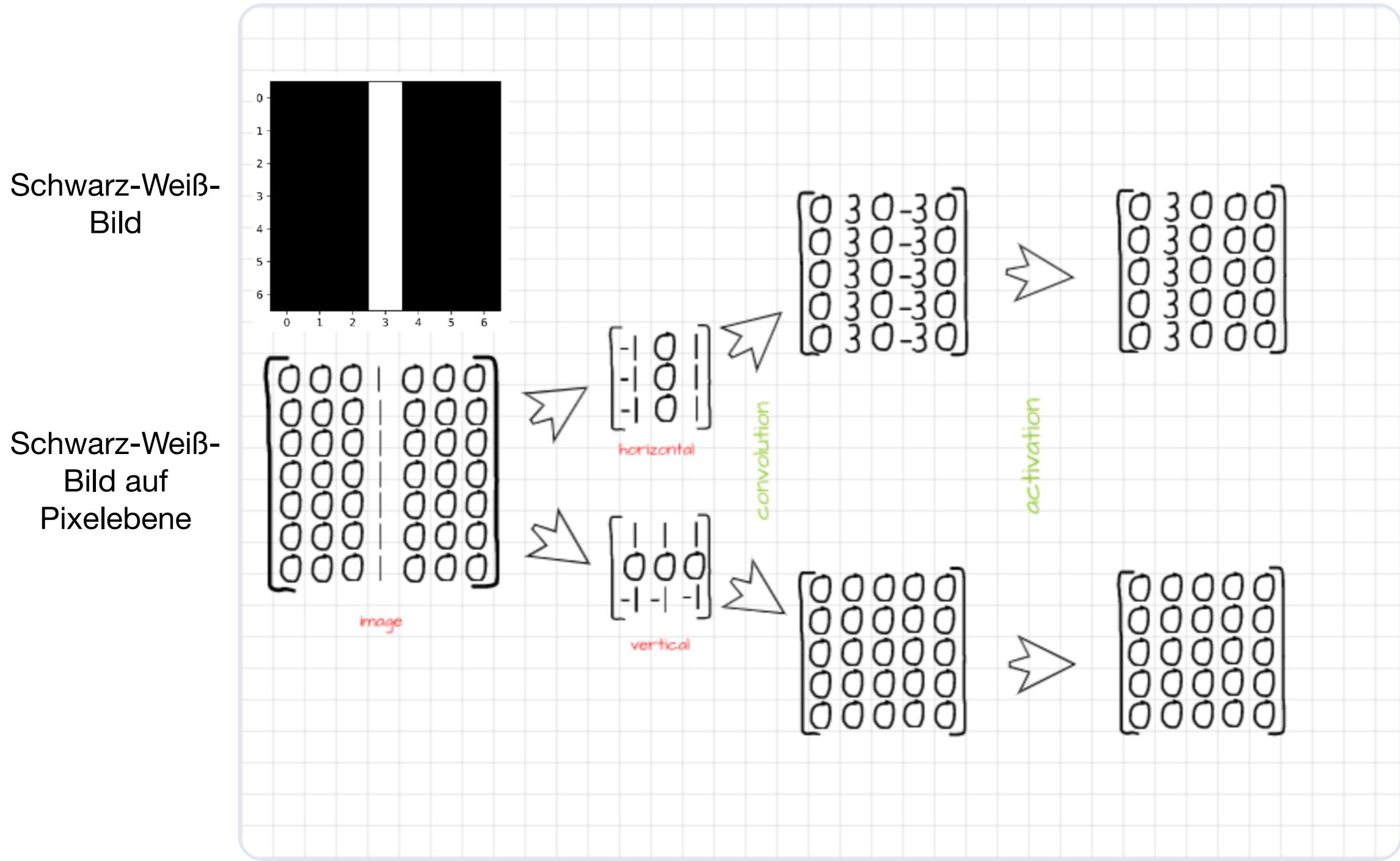


vertikaler Prewitt-Filter

Digital Ocean

Filter / Kernel

Konkrete Kantenerkennung mit dem Prewitt-Filter:



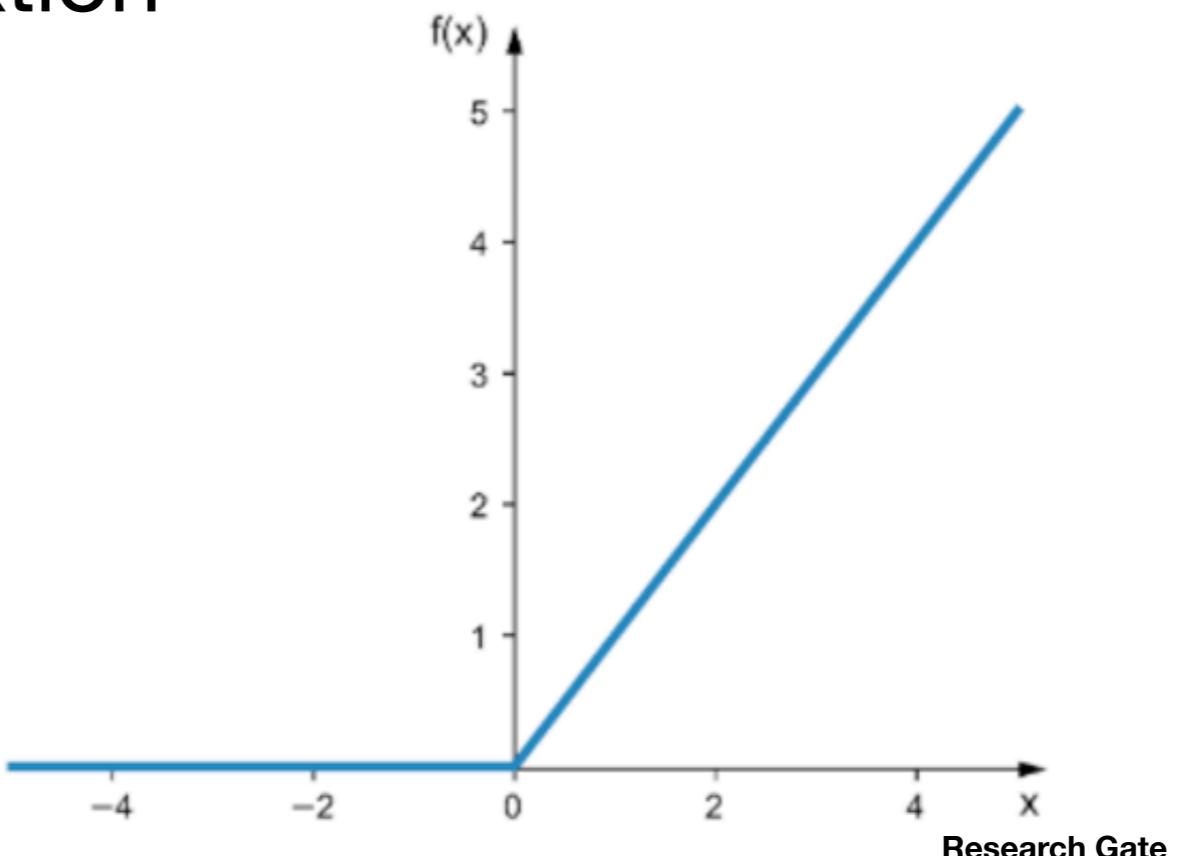
[Digital Ocean](#)

Aktivierungsfunktion

Rectified Linear Unit, auch ReLU

- Aktivierungsfunktion
- schnelle und effiziente Berechnung
- viele Neuronen werden deaktiviert, d.h. auf „0“ gesetzt
- weniger betroffen vom Vanishing-Gradient-Problem als z.B. die Sigmoid-Funktion

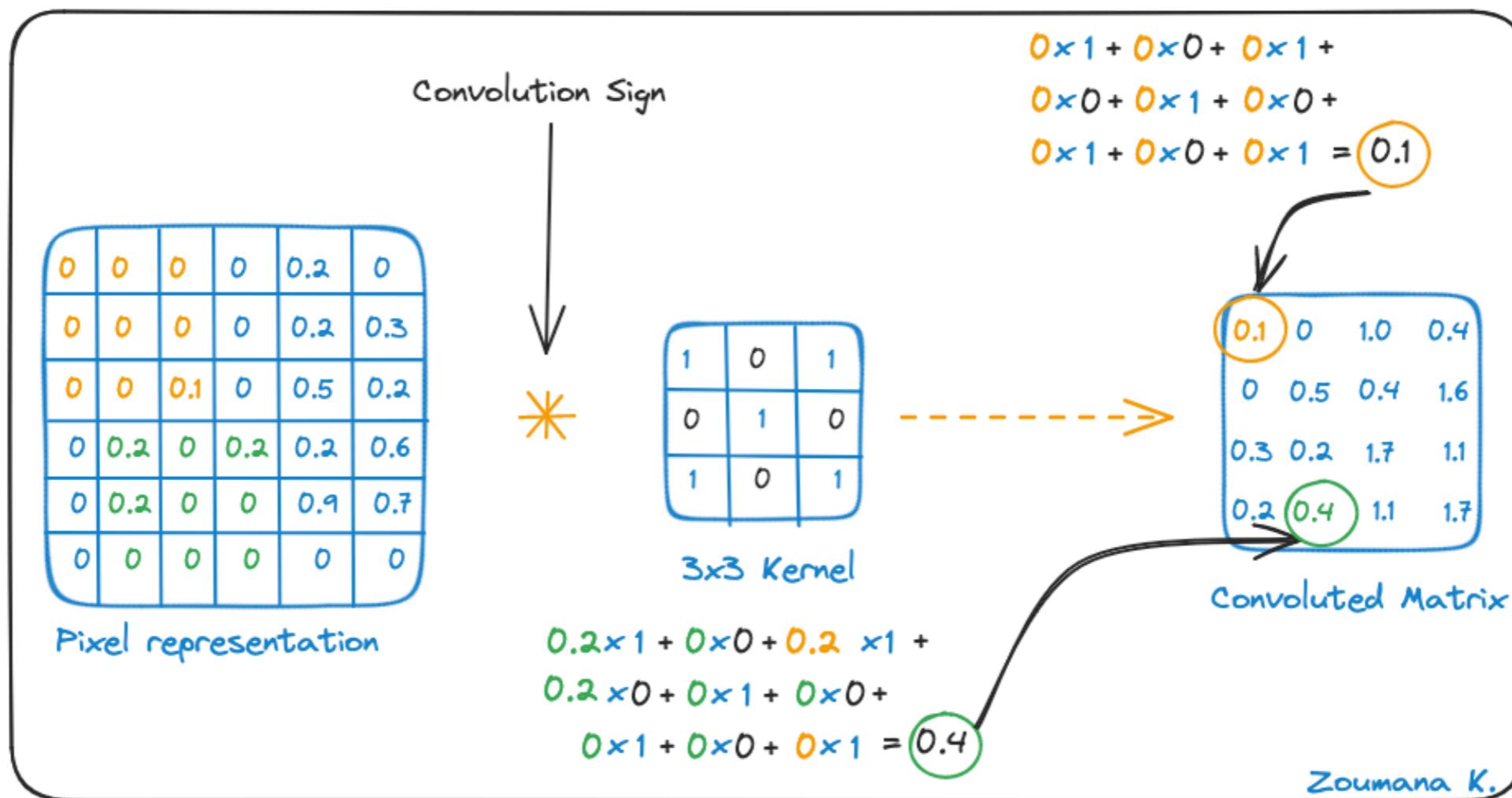
$$f(x) = \text{ReLU}(x) = \max(0, x)$$



[Research Gate](#)

Auswirkungen der Filter auf die Dimension der Ergebnismatrix

Application of the convolution task using
a stride of 1 with 3×3 kernel



Houston, haben wir ein Problem? 🤔

Es könnte sein! 😞

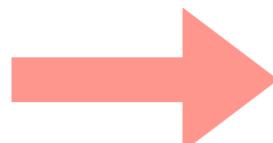
Abhilfe schafft das Padding

Durch das Anwenden von Filtern bei der Faltungsoperation gehen Randinformationen verloren.

Zusätzlich werden Pixel am Rand weniger oft in die Berechnungen einbezogen als die inneren!

Die Dimension der Ausgabematrix schrumpft.

A 28×28 single input channel (grayscale image).



3 x 3 Filter

| | | |
|------|------|------|
| 0.98 | 0.28 | 0.94 |
| 0.71 | 0.05 | 0.56 |
| 0.6 | 0.33 | 0.85 |

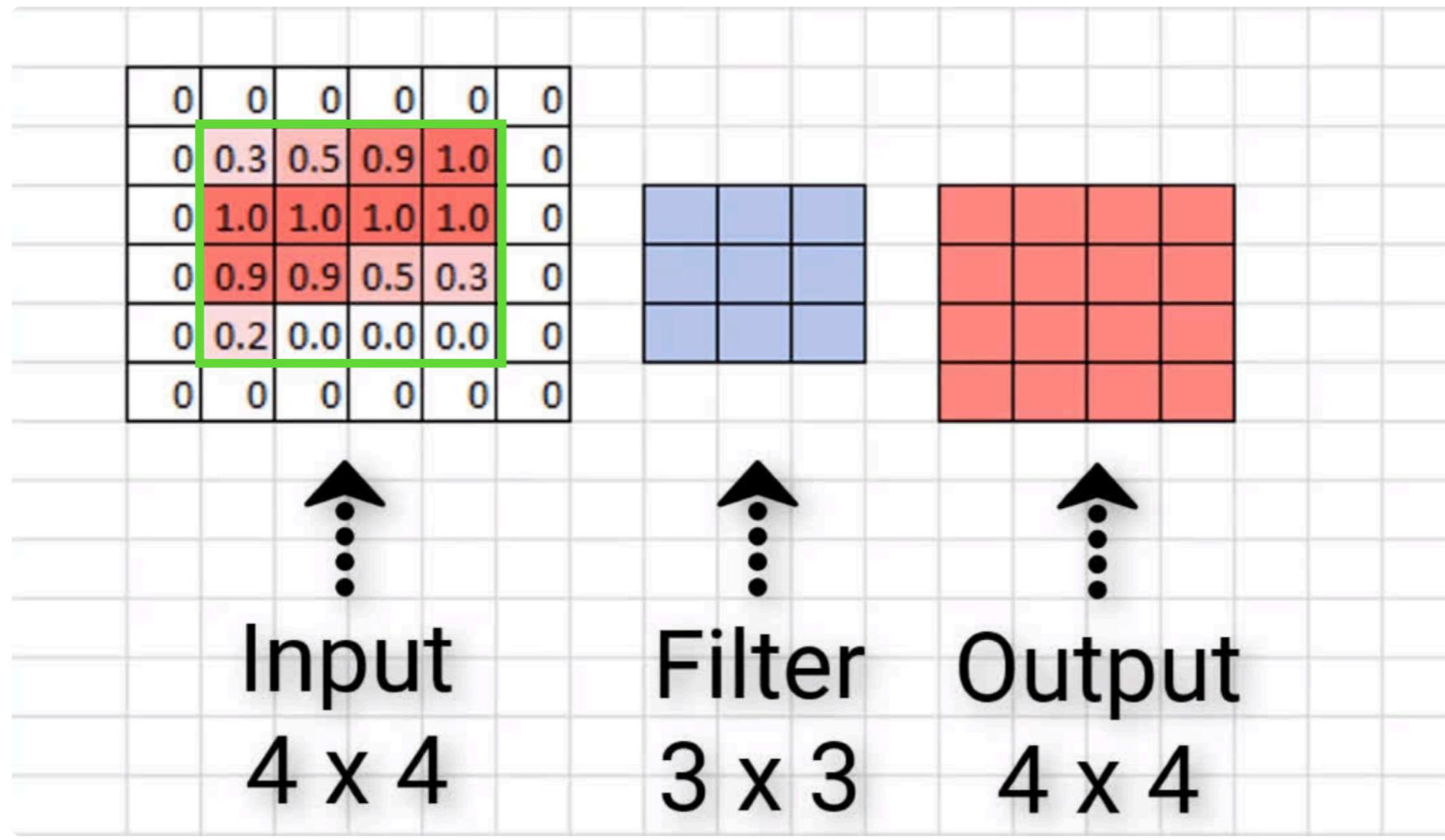
A 26×26 output channel:

| | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 0.3 | 0.6 | 1.2 | 1.4 | 1.6 | 1.6 | 1.6 | 1.9 | 1.9 | 2.2 | 2.3 | 2.1 | 2.0 | 1.7 | 0.9 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 0.5 | 1.2 | 1.8 | 2.6 | 2.7 | 3.0 | 3.0 | 3.0 | 3.4 | 3.5 | 3.8 | 4.0 | 3.7 | 3.6 | 3.2 | 2.3 | 1.5 | 0.5 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 1.1 | 2.1 | 3.2 | 4.2 | 4.4 | 4.7 | 4.7 | 4.5 | 4.2 | 4.0 | 3.8 | 3.9 | 3.9 | 4.1 | 4.5 | 4.7 | 4.1 | 3.1 | 1.5 | 0.5 | 0.0 | 0.0 | 0.0 | |
| 1.1 | 2.0 | 3.1 | 3.6 | 3.3 | 3.2 | 3.2 | 3.1 | 2.9 | 2.7 | 2.5 | 2.5 | 2.5 | 2.7 | 3.0 | 3.3 | 4.4 | 4.1 | 2.9 | 1.4 | 0.3 | 0.0 | 0.0 | |
| 0.9 | 1.4 | 2.1 | 2.2 | 1.8 | 1.7 | 1.7 | 1.5 | 1.1 | 0.8 | 0.5 | 0.5 | 0.5 | 0.8 | 1.3 | 2.4 | 3.7 | 4.5 | 4.0 | 2.4 | 1.0 | 0.0 | 0.0 | |
| 0.1 | 0.3 | 0.3 | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 1.3 | 2.8 | 4.2 | 4.7 | 2.8 | 1.6 | 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 | 0.0 | 0.0 | 0.0 | 0.4 | 1.2 | 2.9 | 3.9 | 5.1 | 3.1 | 2.2 | 0.1 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.4 | 1.0 | 1.3 | 1.6 | 1.9 | 2.4 | 3.7 | 4.4 | 5.2 | 3.8 | 2.5 | 0.7 | 0.0 | 0.0 | |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 0.5 | 1.1 | 1.7 | 2.3 | 2.7 | 3.0 | 3.4 | 3.7 | 4.6 | 4.9 | 5.2 | 4.1 | 2.5 | 1.2 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.7 | 1.3 | 1.9 | 2.6 | 3.2 | 4.0 | 4.4 | 4.8 | 4.4 | 4.2 | 4.5 | 4.8 | 5.2 | 4.5 | 2.7 | 1.6 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.4 | 1.0 | 1.8 | 2.6 | 3.3 | 3.8 | 3.9 | 3.8 | 3.6 | 3.4 | 3.0 | 2.9 | 3.6 | 4.1 | 5.0 | 3.8 | 2.5 | 1.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 1.7 | 3.0 | 3.5 | 3.7 | 3.3 | 3.0 | 2.5 | 2.2 | 1.9 | 1.3 | 1.3 | 2.4 | 3.3 | 4.8 | 3.4 | 2.3 | 0.6 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.9 | 2.0 | 2.7 | 3.2 | 2.6 | 1.8 | 1.3 | 0.7 | 0.4 | 0.1 | 0.0 | 0.4 | 2.2 | 3.3 | 4.6 | 3.0 | 2.0 | 0.2 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 1.4 | 1.6 | 1.7 | 0.7 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 2.5 | 3.7 | 4.2 | 2.6 | 1.5 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.5 | 0.2 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 1.7 | 3.3 | 4.0 | 3.6 | 2.2 | 0.8 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 1.3 | 2.3 | 4.0 | 3.9 | 2.8 | 1.6 | 0.2 | 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 | 0.0 | 0.3 | 2.3 | 3.1 | 4.5 | 3.4 | 2.0 | 0.8 | 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 | 0.0 | 0.0 | 0.0 | 0.9 | 2.6 | 3.4 | 3.8 | 2.5 | 1.2 | 0.2 | 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 | 0.0 | 0.0 | 1.2 | 2.0 | 2.8 | 4.1 | 1.5 | 0.3 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.3 | 2.0 | 1.3 | 0.6 | 0.0 | 0.0 | |

DeepLizard

Mit Hilfe des sogenannten Padding (auch Zero-Padding) lässt sich dies umgehen!

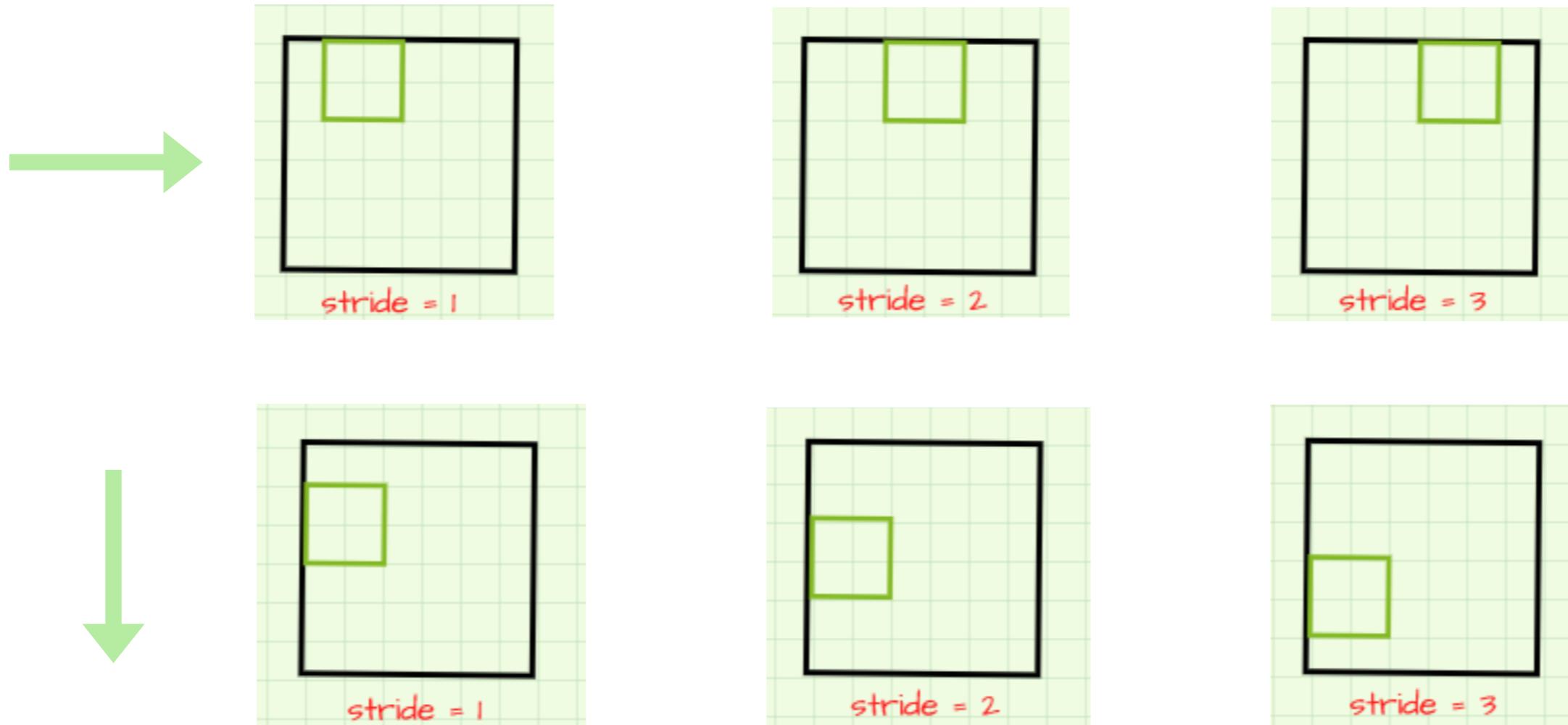
Abhilfe schafft das Zero-Padding



Same-Padding: Fügt Pixel der Eingabe hinzu, sodass die Ausgabe die gleiche Größe hat und die Randpixel berücksichtigt werden.

Valid-Padding: Fügt keine Pixel der Eingabe hinzu, sodass die Ausgabe verkleinert und Randpixel ignoriert werden.

Schrittweite / Stride



[Digital Ocean](#)

Die Schrittweite definiert wie viele Einheiten der Filter gleitet!
Sie hat Einfluss auf die Dimension der Ergebnismatrix, je größer der Schritt desto kleiner die Dimension im Ergebnis!

Ergebnismatrix

Die Dimension einer zweidimensionalen Ergebnismatrix O wird durch die folgenden Eigenschaften bestimmt:

$$O = \left[\frac{n - f + 2p}{s} \right] + 1$$

O = Output size

n = Input size

f = Kernel size

p = Padding

s = Stride



Pooling

Die Pooling-Schicht befindet sich in der Regel hinter einer oder mehreren Konvolutionalen Schicht(en) und verarbeitet die Ergebnisse dieser.

Intention:

Reduzierung der Dimensionalität (Downsampling), dem Rechenaufwand und der Parameter bei **Bewahrung** der wesentlichen Merkmale (Informationen)!

Bewährte Techniken: Max-Pooling, Average-Pooling und Min-Pooling

Die Filter werden, im Gegensatz zur konvolutionalen Schicht, nicht überlappend angewendet.



Pooling

Ergebnis der konvolutionalen Schicht

A 26 x 26 output channel:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 0.0 | 0.3 | 0.6 | 1.2 | 1.4 | 1.6 | 1.6 | 1.6 | 1.9 | 1.9 | 2.2 | 2.3 | 2.1 | 2.0 | 1.7 | 0.9 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 0.5 | 1.2 | 1.8 | 2.6 | 2.7 | 3.0 | 3.0 | 3.0 | 3.4 | 3.4 | 3.8 | 4.0 | 3.7 | 3.6 | 3.2 | 2.3 | 1.5 | 0.5 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 1.1 | 2.1 | 3.2 | 4.2 | 4.4 | 4.7 | 4.5 | 4.2 | 4.0 | 3.8 | 3.9 | 3.9 | 4.1 | 4.5 | 4.7 | 4.1 | 3.1 | 1.5 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 1.1 | 2.0 | 3.1 | 3.6 | 3.3 | 3.2 | 3.1 | 2.9 | 2.7 | 2.5 | 2.5 | 2.5 | 2.7 | 3.0 | 3.9 | 4.4 | 4.1 | 2.9 | 1.4 | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 0.9 | 1.4 | 2.1 | 2.2 | 1.8 | 1.7 | 1.7 | 1.5 | 1.1 | 0.8 | 0.5 | 0.5 | 0.5 | 0.8 | 1.3 | 2.4 | 3.7 | 4.5 | 4.0 | 2.4 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.1 | 0.3 | 0.3 | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 1.3 | 2.8 | 4.2 | 4.7 | 2.8 | 1.6 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.4 | 1.2 | 2.9 | 3.9 | 5.1 | 3.1 | 2.2 | 0.1 | 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 | 0.0 | 0.1 | 0.4 | 1.0 | 1.3 | 1.6 | 1.9 | 2.4 | 3.7 | 4.4 | 5.2 | 3.8 | 2.5 | 0.7 | 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 | 0.2 | 0.5 | 1.1 | 1.7 | 2.3 | 2.7 | 3.0 | 3.4 | 3.7 | 4.6 | 4.9 | 5.2 | 4.1 | 2.5 | 1.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.7 | 1.3 | 1.9 | 2.6 | 3.2 | 4.0 | 4.4 | 4.8 | 4.4 | 4.2 | 4.5 | 4.8 | 5.2 | 4.5 | 2.7 | 1.6 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.4 | 1.0 | 1.8 | 2.6 | 3.3 | 3.8 | 3.9 | 3.8 | 3.6 | 3.4 | 3.0 | 2.9 | 3.6 | 4.1 | 5.0 | 3.8 | 2.5 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 1.7 | 3.0 | 3.5 | 3.7 | 3.3 | 3.0 | 2.5 | 2.2 | 1.9 | 1.3 | 1.3 | 2.4 | 3.3 | 4.8 | 3.4 | 2.3 | 0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.9 | 2.0 | 2.7 | 3.2 | 2.6 | 1.8 | 1.3 | 0.7 | 0.4 | 0.1 | 0.0 | 0.4 | 2.2 | 3.3 | 4.6 | 3.0 | 2.0 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 1.4 | 1.6 | 1.7 | 0.7 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 1.7 | 3.3 | 4.0 | 3.6 | 2.2 | 0.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.5 | 0.2 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.3 | 2.3 | 4.0 | 3.9 | 2.8 | 1.6 | 0.2 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.3 | 2.3 | 3.1 | 4.5 | 3.4 | 2.0 | 0.8 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.9 | 2.6 | 3.4 | 3.8 | 2.5 | 1.2 | 0.2 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.2 | 2.0 | 2.8 | 2.4 | 1.5 | 0.3 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.3 | 2.0 | 1.3 | 0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

| | |
|-----|-----|
| 1,1 | 2,1 |
| 1,1 | 2,0 |

Ergebnis nach dem Max-Pooling mit einem 2x2 Filter und einer Schrittweite 2:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 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 | 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 | |
| 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 | 0.0 | 0.0 | 0.0 | 0.3 | 0.6 | 0.7 | 0.4 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.2 | 2.6 | 3.0 | 3.4 | 3.8 | 4.0 | 3.6 | 2.3 | 0.5 | 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 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.1 | 4.2 | 4.7 | 4.2 | 3.9 | 4.1 | 4.7 | 4.4 | 2.9 | 0.3 | 0.0 | 0.0 |
| 1.4 | 2.2 | 1.8 | 1.7 | 1.1 | 0.5 | 0.8 | 2.4 | 4.5 | 4.7 | 4.1 | 4.7 | 4.4 | 2.4 | 4.5 | 4.7 | 1.6 | 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 | 0.1 | 1.3 | 2.6 | 4.0 | 4.8 | 4.4 | 4.9 | 5.2 | 2.7 | 0.0 | 0.0 | | | | | | | | | | | | | | | |

Pooling

Max-Pooling vs. Average-Pooling vs. Min-Pooling

Max-Pooling

- Hervorheben der hellsten Pixel
- Nützlich, wenn der Hintergrund des Bildes dunkel ist und die helleren Pixel die Information enthalten
- Beispiel: MNIST-Datensatz → die Ziffern sind weiß und der Hintergrund ist schwarz

Average-Pooling

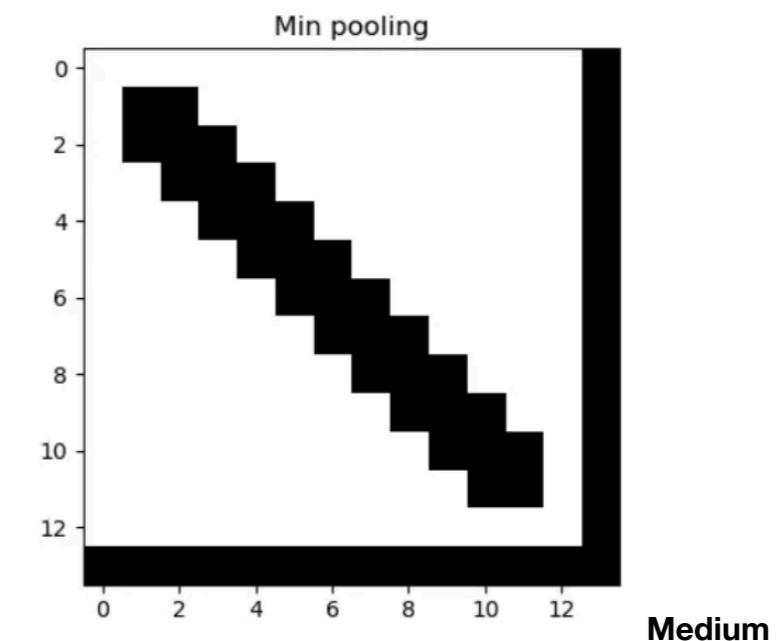
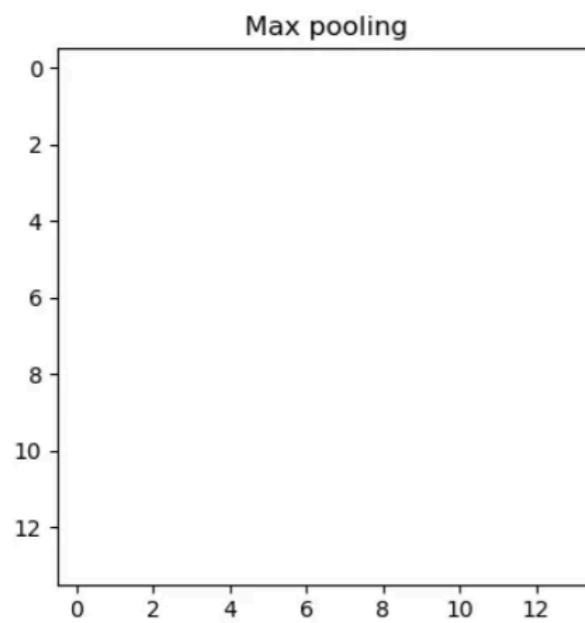
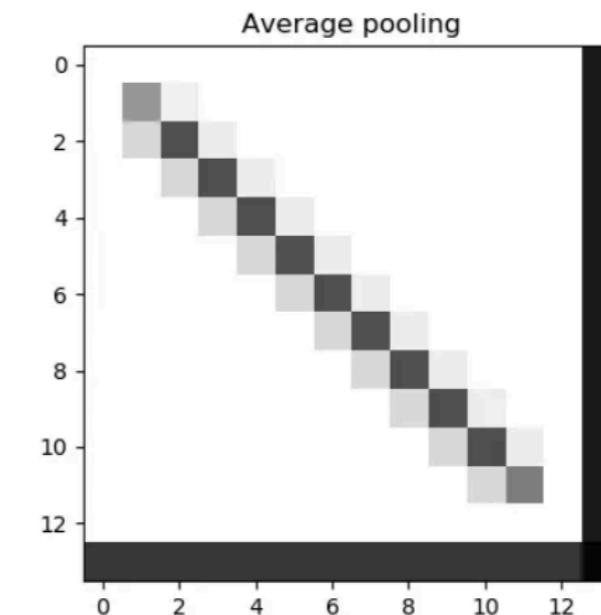
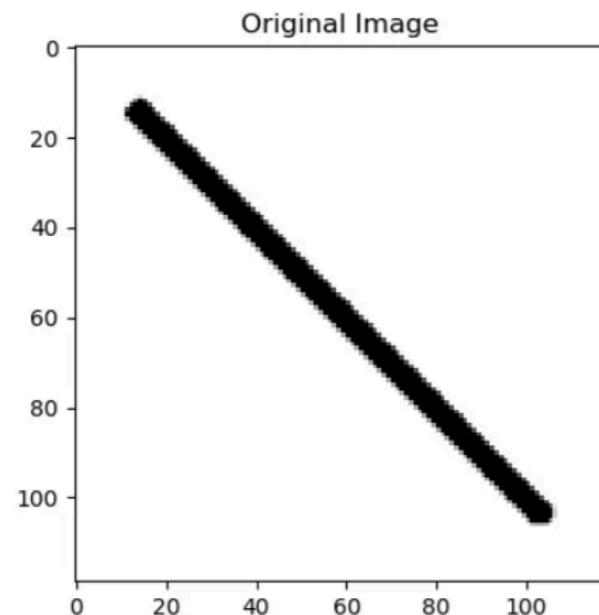
- Glättet das Bild
- scharfe Merkmale werden u.U. nicht erkannt

Min-Pooling

- Gegenteil von Max-Pooling
- Hervorheben der dunkleren Pixel
- Nützlich, wenn der Hintergrund des Bildes hell ist und die dunkleren Pixel die Information enthalten

Pooling

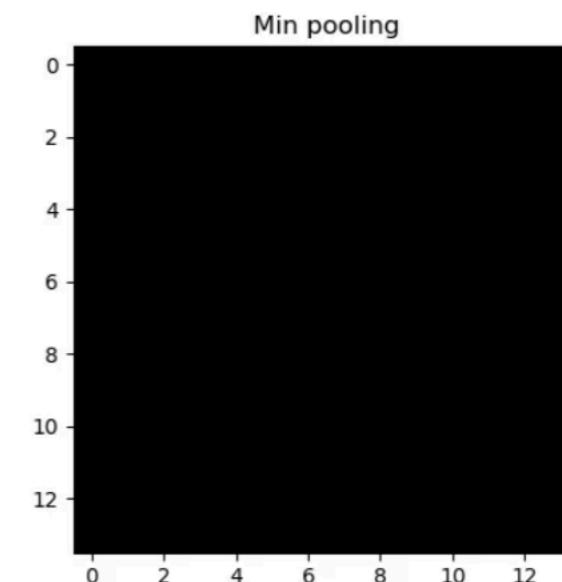
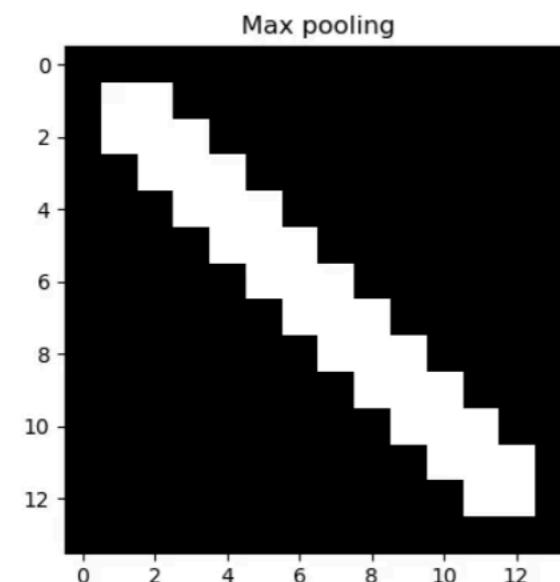
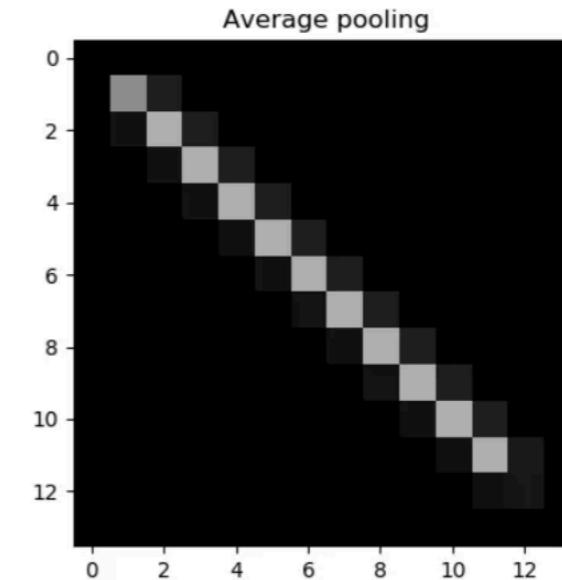
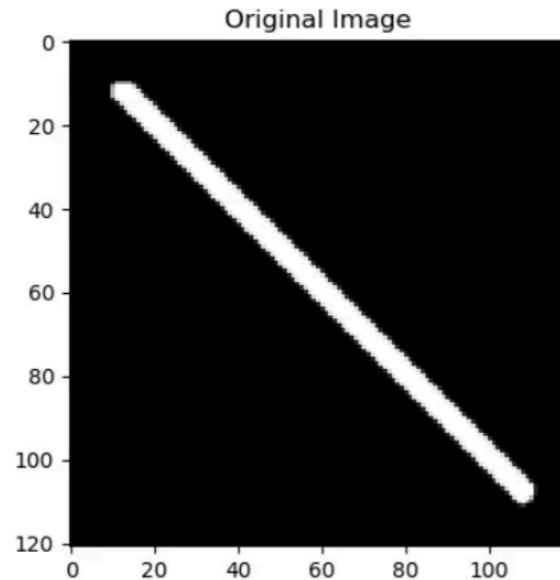
Max-Pooling vs. Average-Pooling vs. Min-Pooling



Min-Pooling erzielt ein besseres Ergebnis bei hellem Hintergrund und dunklen Objekten

Pooling

Max-Pooling vs. Average-Pooling vs. Min-Pooling



Medium

Max-Pooling erzielt ein besseres Ergebnis bei dunklem Hintergrund und hellen Objekten

Pooling

Min- und Max-Pooling werden verwendet, um die auffälligsten Merkmale zu extrahieren

Weniger auffällige Merkmale werden ignoriert, wie z.B. Rauschen oder „nicht relevante“ Information.

Average-Pooling wird verwendet, wenn feine Details und/oder Texturen wichtig sind.

Merkmalsextraktion

Die konvolutionalen und Pooling-Schichten haben die Aufgabe Merkmale zu extrahieren

Durch die hintereinander geschalteten Faltungen erhält man

- eine hierarchische Darstellung des Eingabebildes
- den Effekt, dass je tiefer die Ebenen im CNN, desto komplexer und abstrakter sind die Merkmale, die durch die Filter dieser Ebene extrahiert werden können:
 - vorderste Ebenen: Kanten oder Texturen
 - tiefere Ebenen: Formen oder Objekte

Vollständig verbundene Schicht

Die vollständig verbundene Schicht ist für die Klassifikation der extrahierten Merkmale aus den vorhergehenden Schichten zuständig

Dies erfolgt mit Hilfe von klassischen Feed-Forward-Netzen, deren Schichten voll vermascht sind.

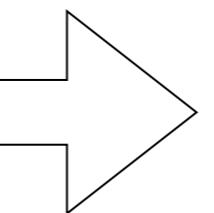
Die Ausgabe / Output der letzten Pooling-Schicht(en) dient hierfür als Eingabe / Input.

Vollständig verbundene Schicht

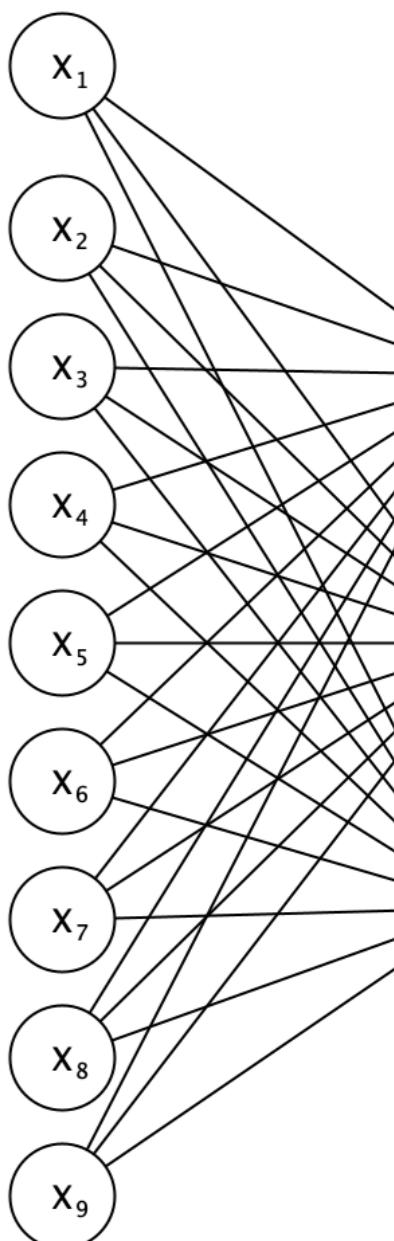
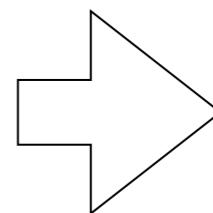
Das Format der Ausgabe / Output muss hierfür angepasst werden durch das sogenannte Flattening/ Abflachen:

Pooled Feature Map /
Merkmalskarte nach dem Pooling

| | | |
|---|---|---|
| 2 | 3 | 4 |
| 1 | 0 | 1 |
| 3 | 1 | 1 |

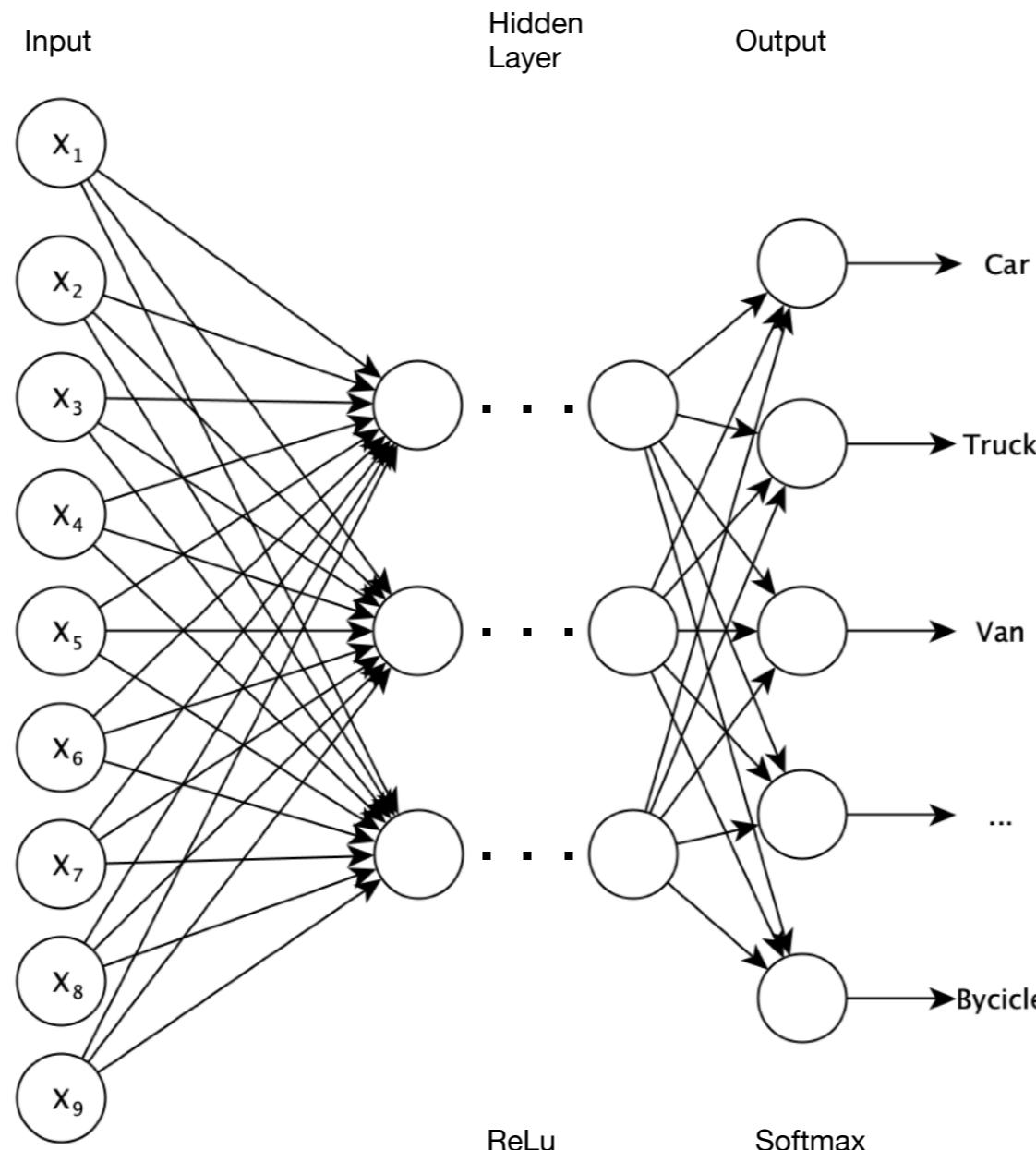


| | | |
|---|---|---|
| 2 | 3 | 4 |
| 1 | 0 | 1 |
| 3 | 1 | 1 |



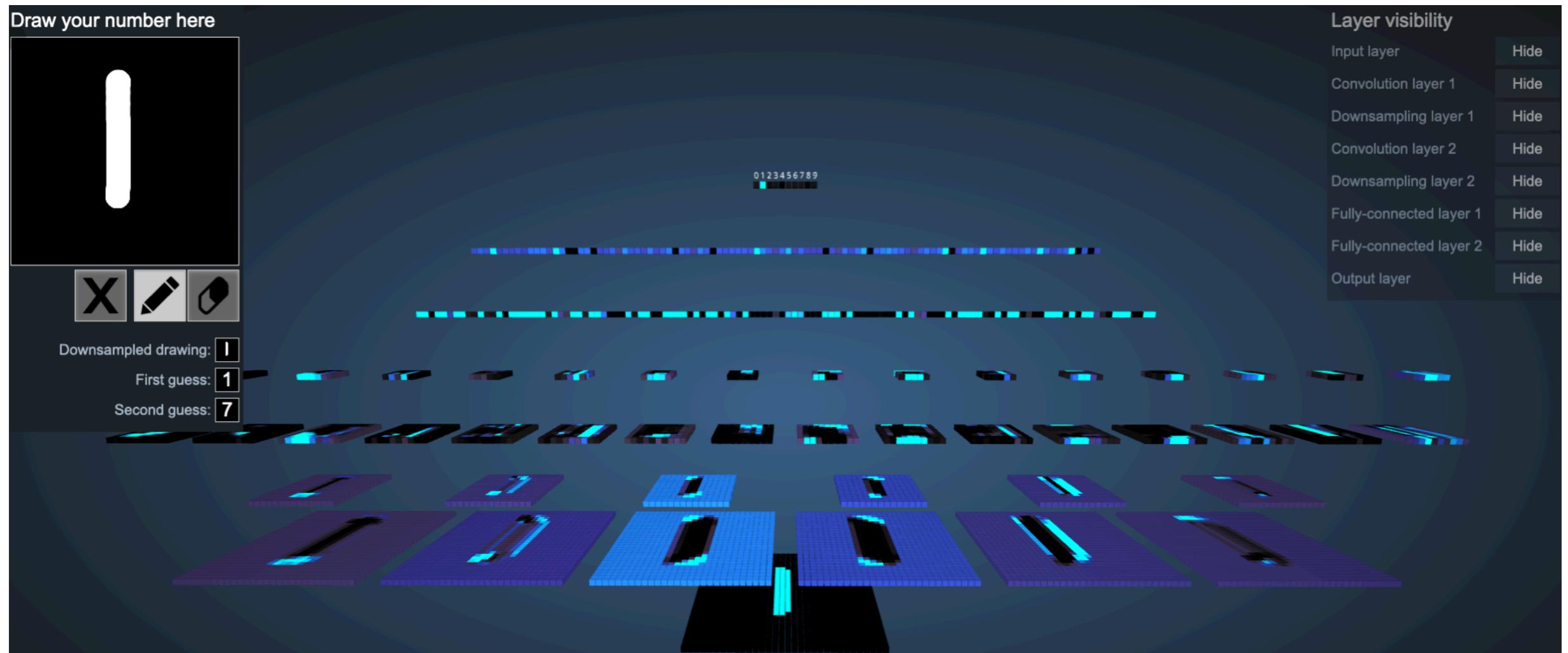
Vollständig verbundene Schicht

Anschließend erfolgt eine Klassifikation mit einem voll vermaschten Neuronalen Netzwerk (Feed Forward):



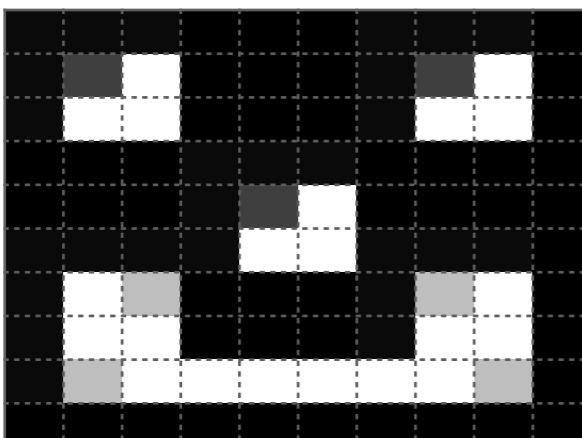
Playground für CNN

Adam Harley, Postdoc Stanford University



Hausaufgabe 05

1. Berechnen Sie die Größe der Dimension der Ausgabe eines Convolutional Layers für die folgenden Parameter:
 - Größe der Eingabe: 64×48
 - Filtergröße: 3×3
 - Stride: 1
 - Padding: „valid“
2. Berechnen Sie mit Hilfe des Max-Poolings das Ergebnis des Pooling-Layers bei folgendem gegebenen Convolutional Layer. Die Schrittweite soll 2 betragen und die Filtergröße 2×2 .



| | | | | | | | | | |
|----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| 10 | 10 | 10 | 0 | 0 | 0 | 10 | 10 | 10 | 0 |
| 10 | 63 | 255 | 0 | 0 | 0 | 10 | 63 | 255 | 0 |
| 10 | 255 | 255 | 0 | 0 | 0 | 10 | 255 | 255 | 0 |
| 0 | 0 | 0 | 10 | 10 | 10 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 10 | 63 | 255 | 0 | 0 | 0 | 0 |
| 10 | 10 | 10 | 10 | 255 | 255 | 10 | 10 | 10 | 0 |
| 10 | 255 | 190 | 0 | 0 | 0 | 10 | 190 | 255 | 0 |
| 10 | 255 | 255 | 0 | 0 | 0 | 10 | 255 | 255 | 0 |
| 10 | 190 | 255 | 255 | 255 | 255 | 255 | 255 | 190 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

3. Was ist die Folge, wenn in einem Künstlichen Neuronalen Netz keine Aktivierungsfunktionen verwendet werden? Recherchieren Sie nach drei gängigen Aktivierungsfunktionen, beschreiben Sie deren Verhalten und in welchen Anwendungsfällen sie eingesetzt werden und warum.