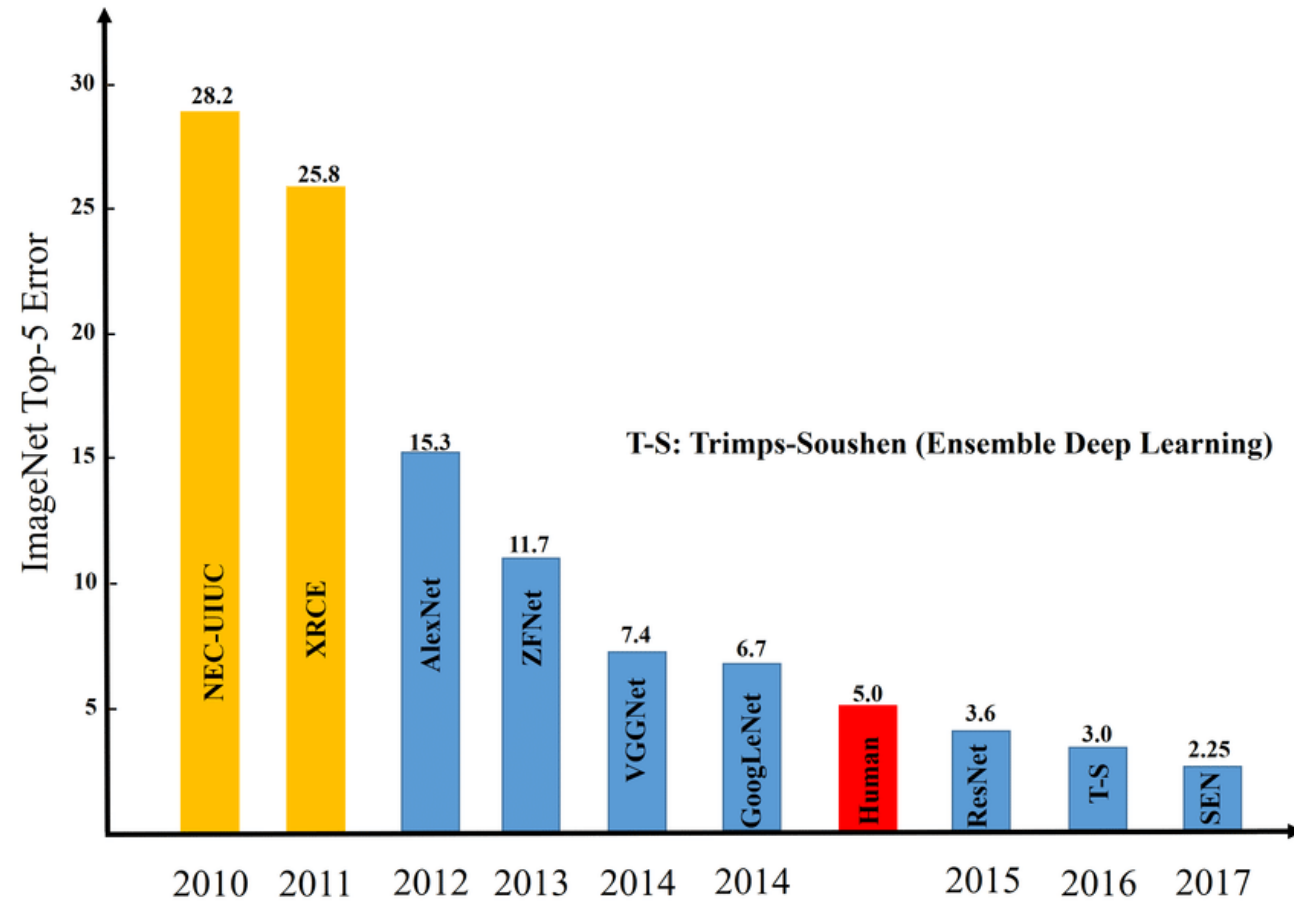


Vor Convolutional Neural Networks ...



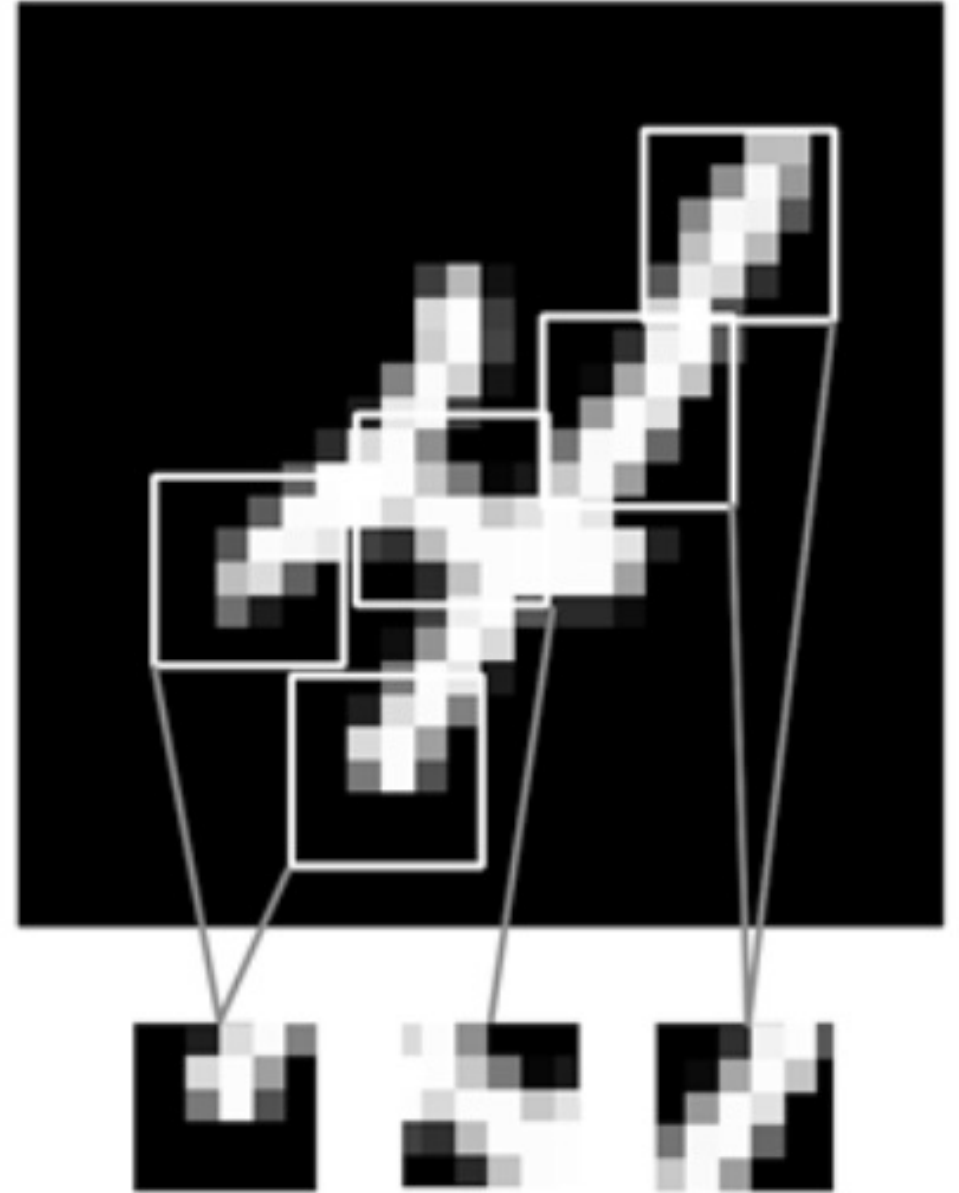
IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

ConvNets in blau

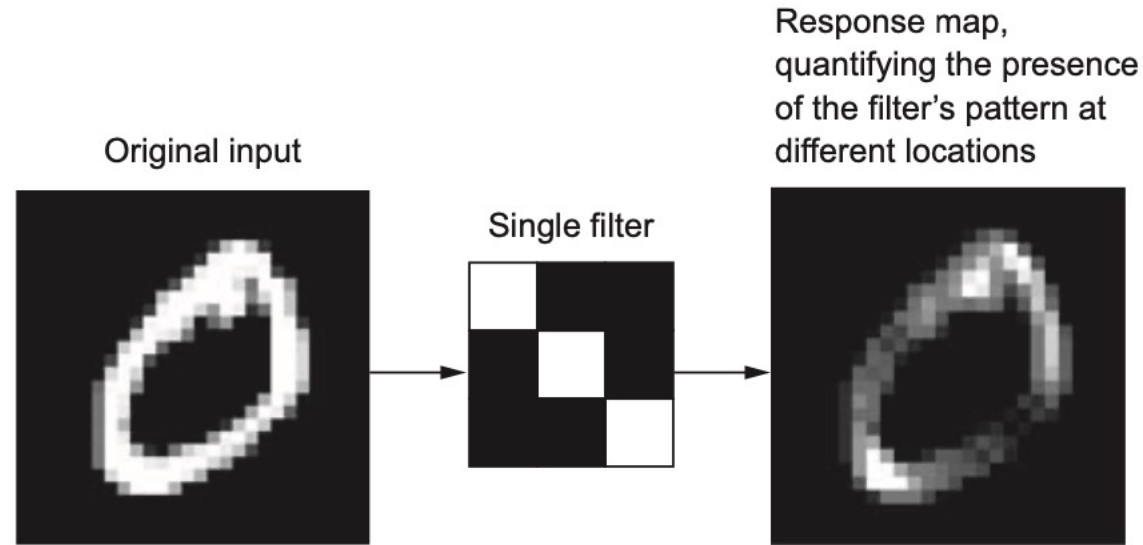


Warum ConvNets?

- Bilder und Buchstaben sind translationsinvariant
- Bilder können in lokale Muster untergebrochen werden
- Dense Layers müssen lokale Muster überall neu lernen

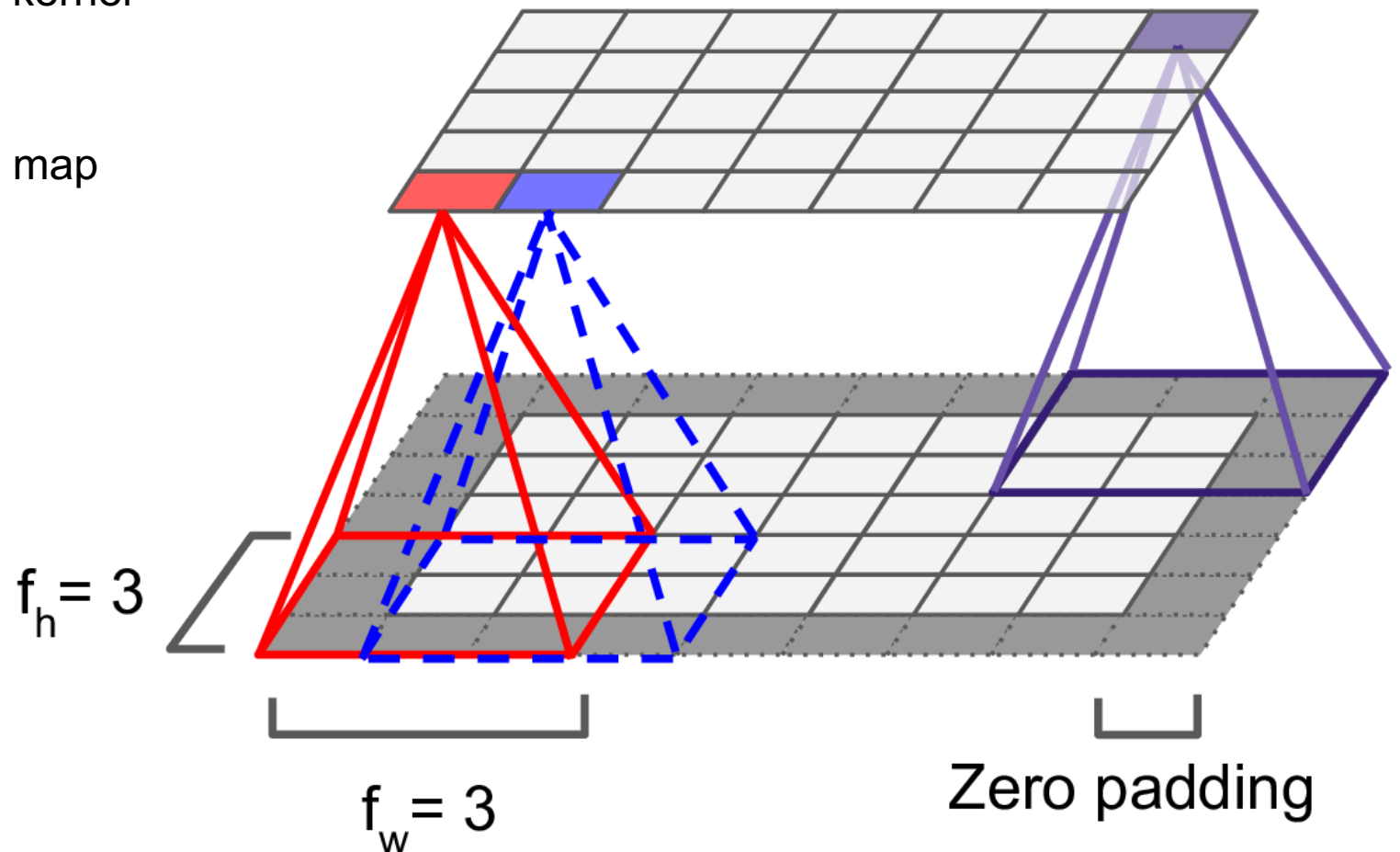


ConvNets sind aus lokalen, verschiebbaren Filtern aufgebaut.

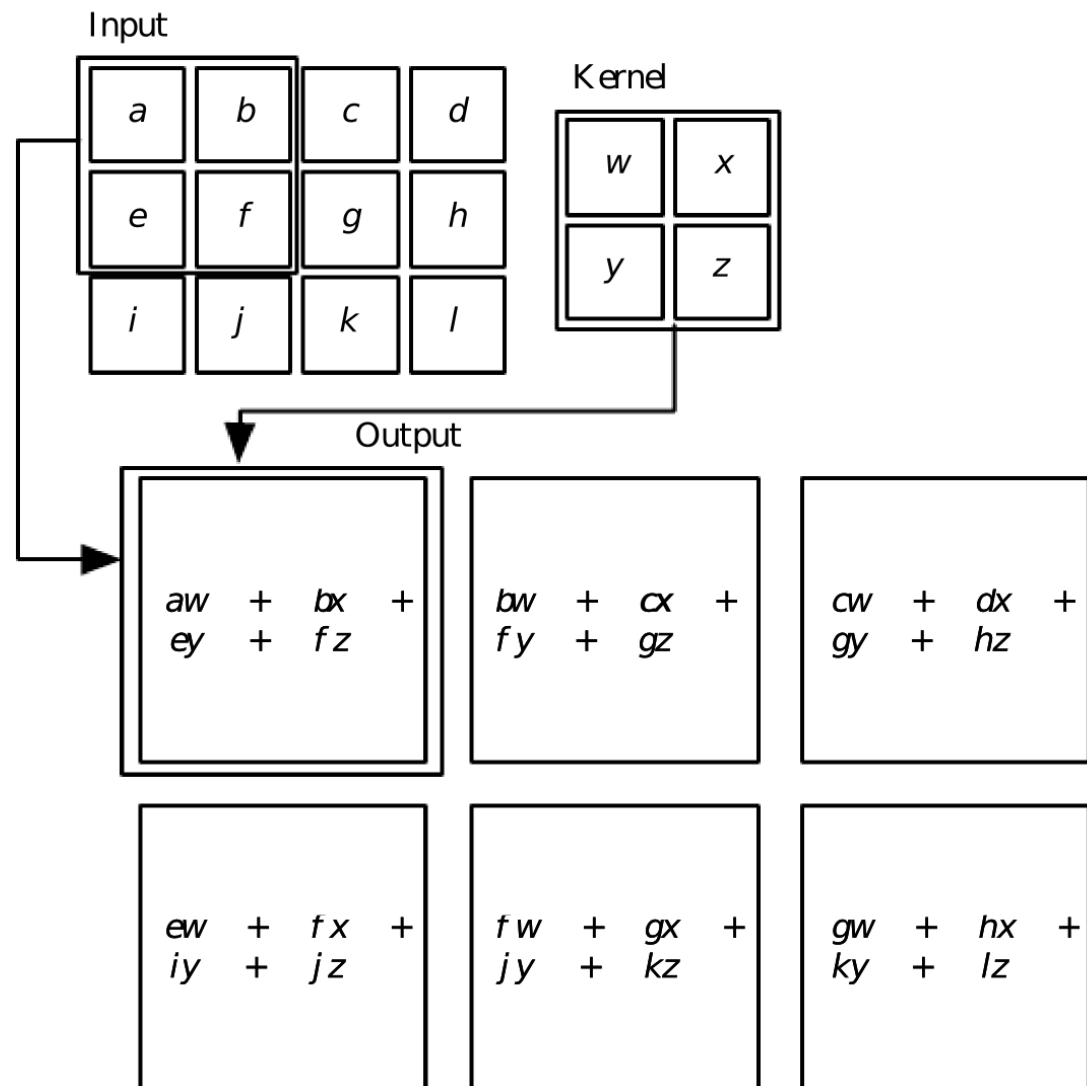


ConvNets sind aus lokalen, verschiebbaren Filtern aufgebaut

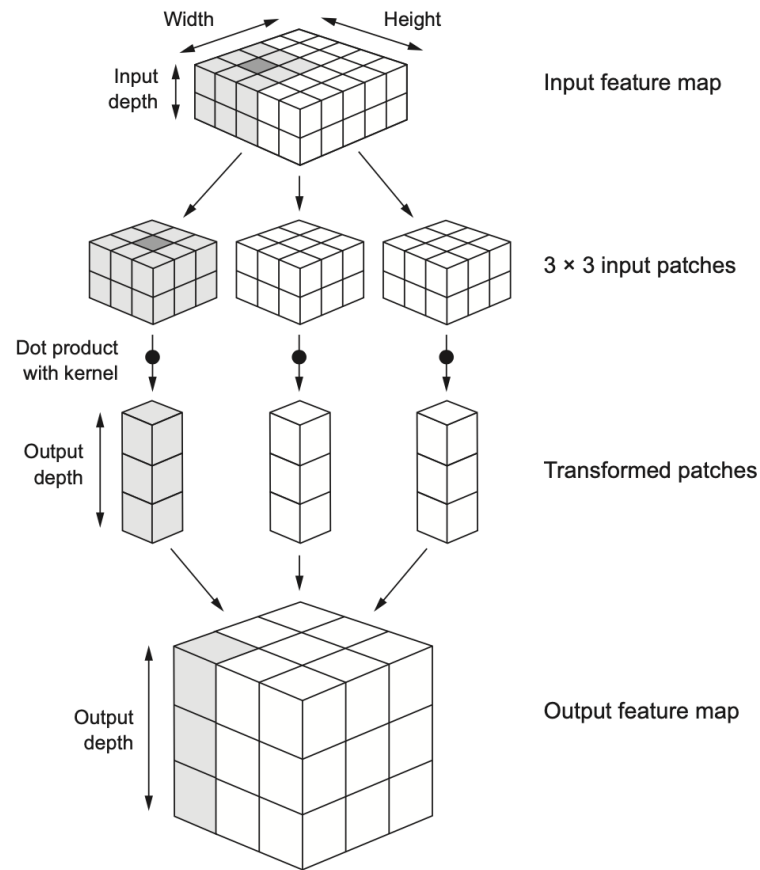
- 3x3 Patches von Gewichten heißen kernel oder filter
- Output eines kernels ist eine feature map
- Am Rand kann man zero padding verwenden oder kein Padding



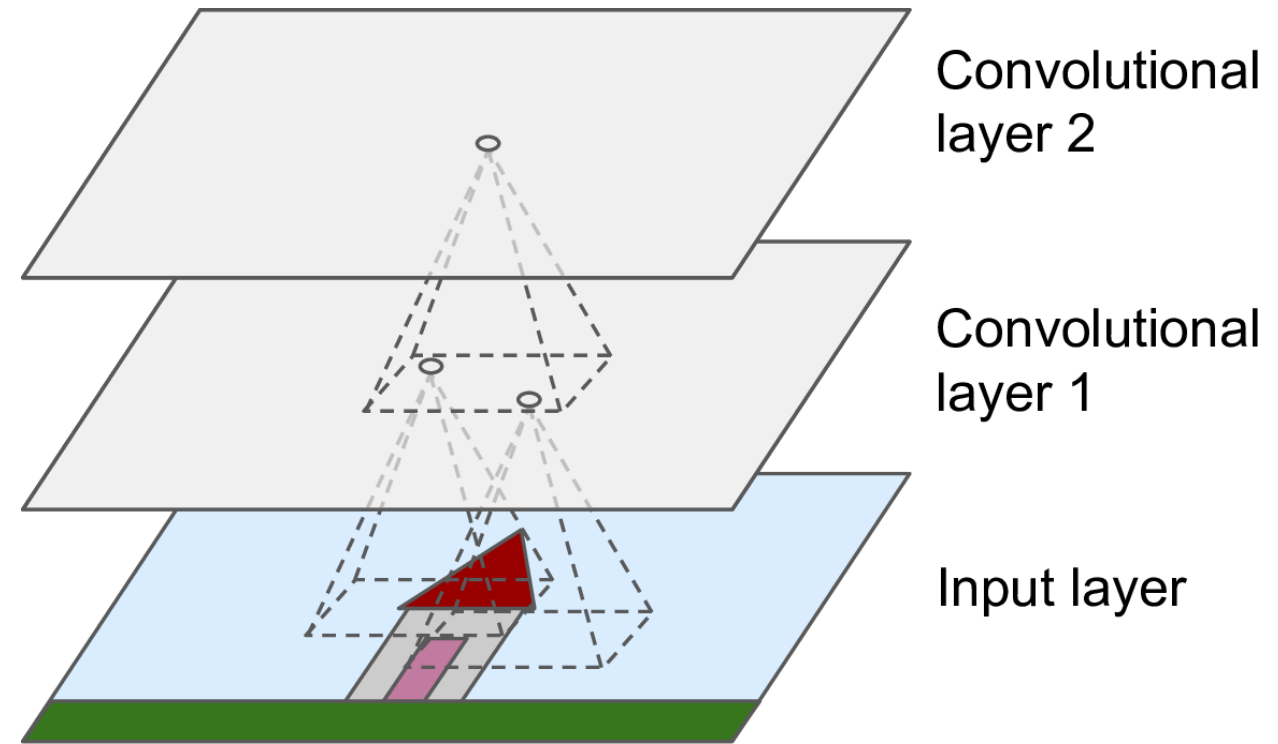
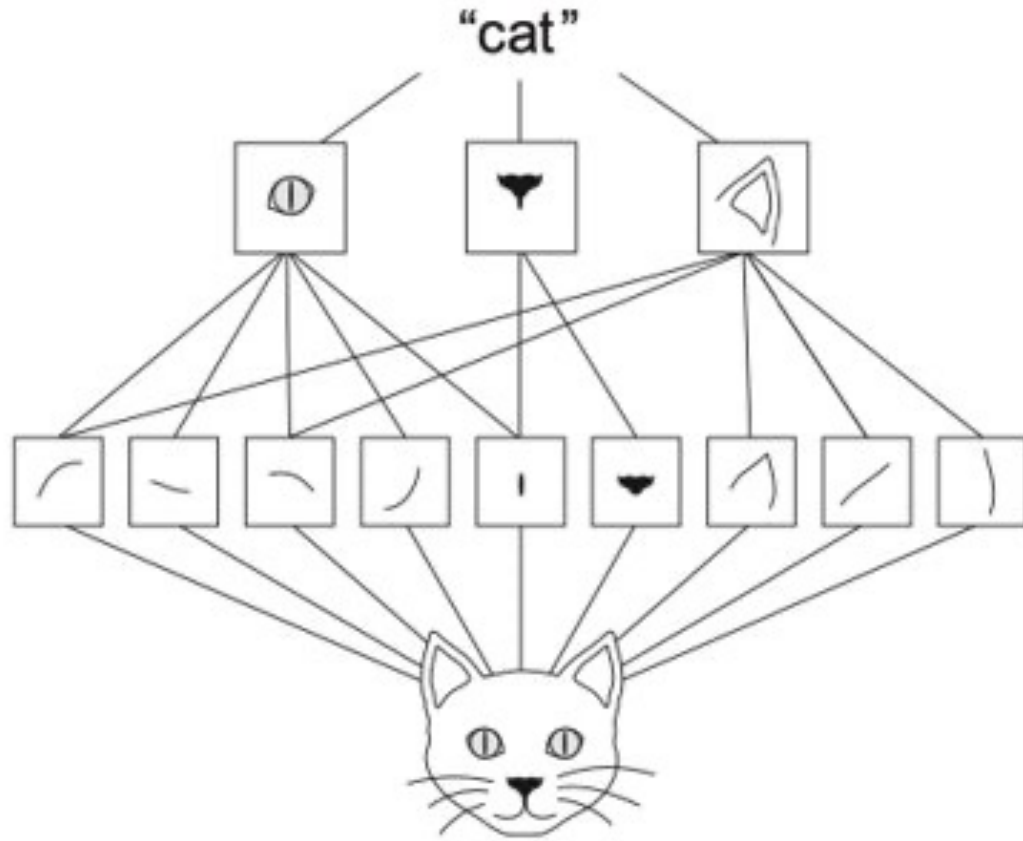
Convolution



Input und Output eines Convolutional-Layers



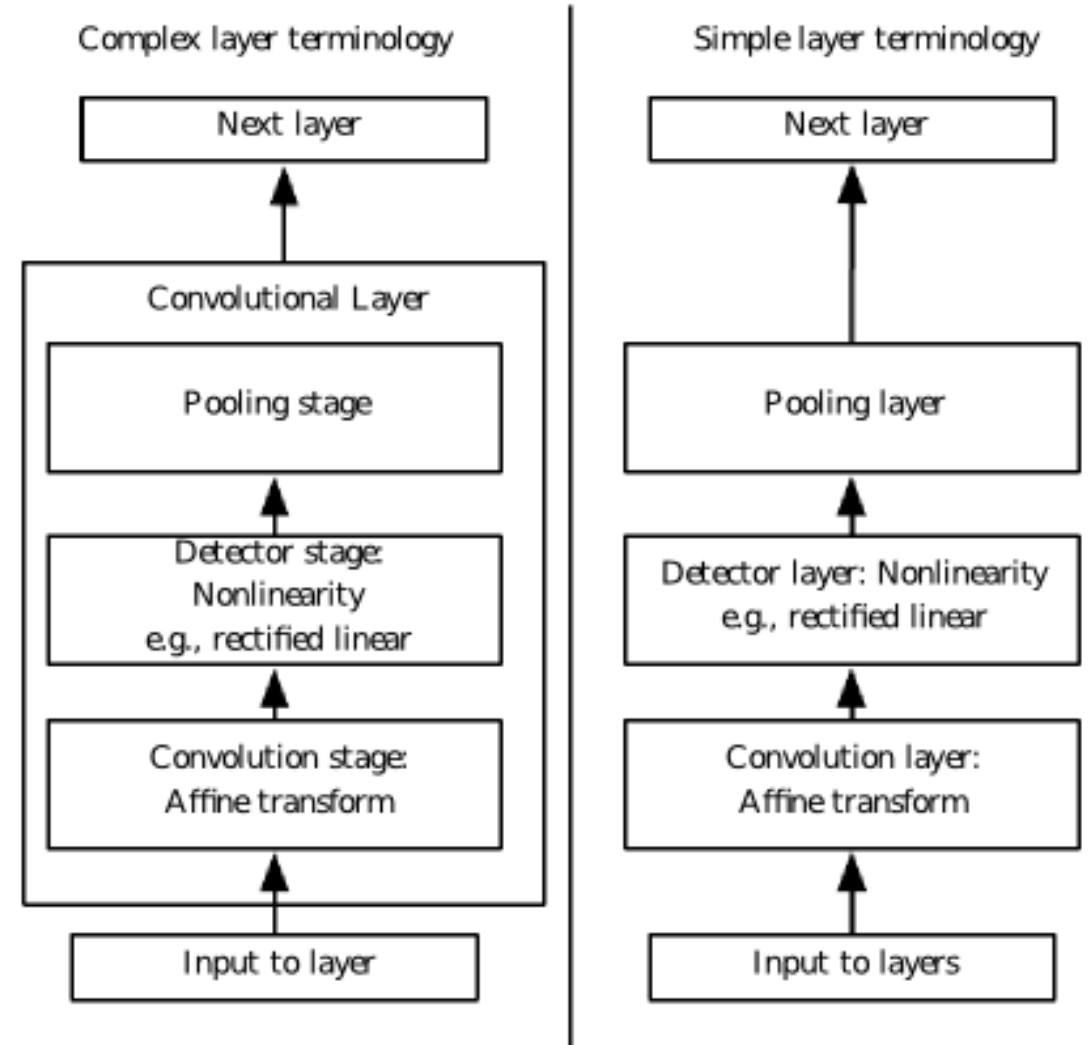
Hierarchie lokaler Muster



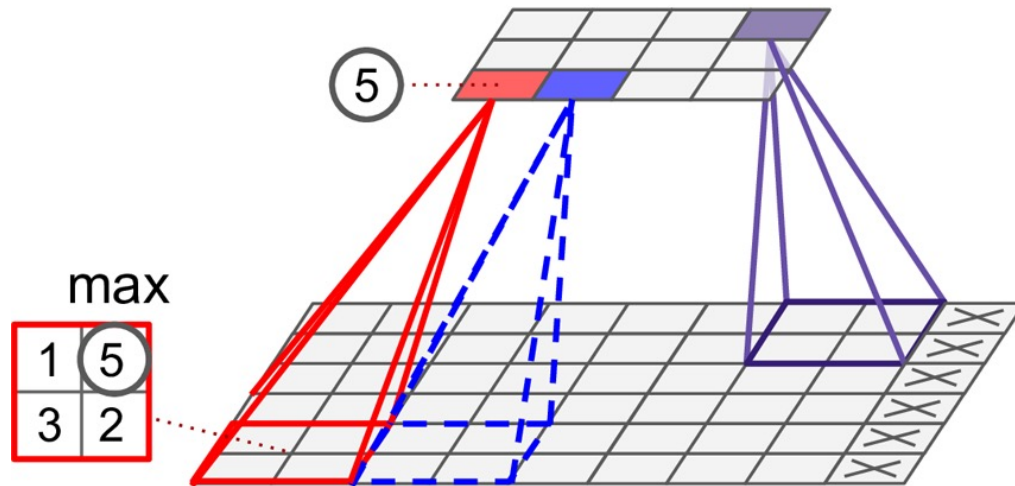
ConvNets sind aus lokalen, verschiebbaren Filtern aufgebaut

Convolutional Layers bestehen aus 3 Teilen:

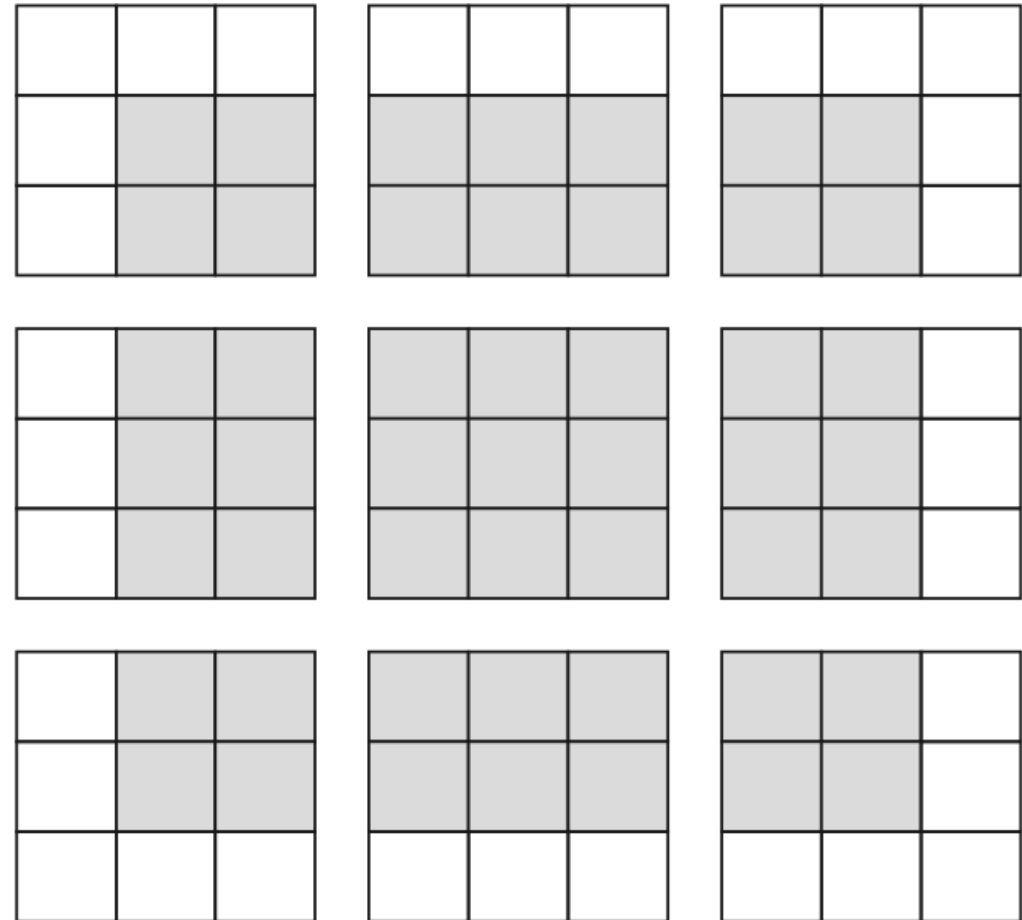
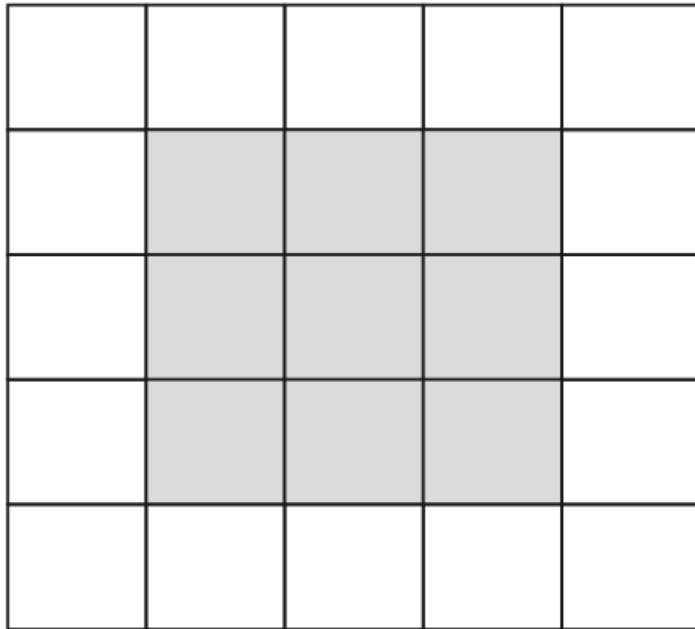
1. Convolution
2. Detector (Aktivierungsfunktion)
3. Pooling Stage



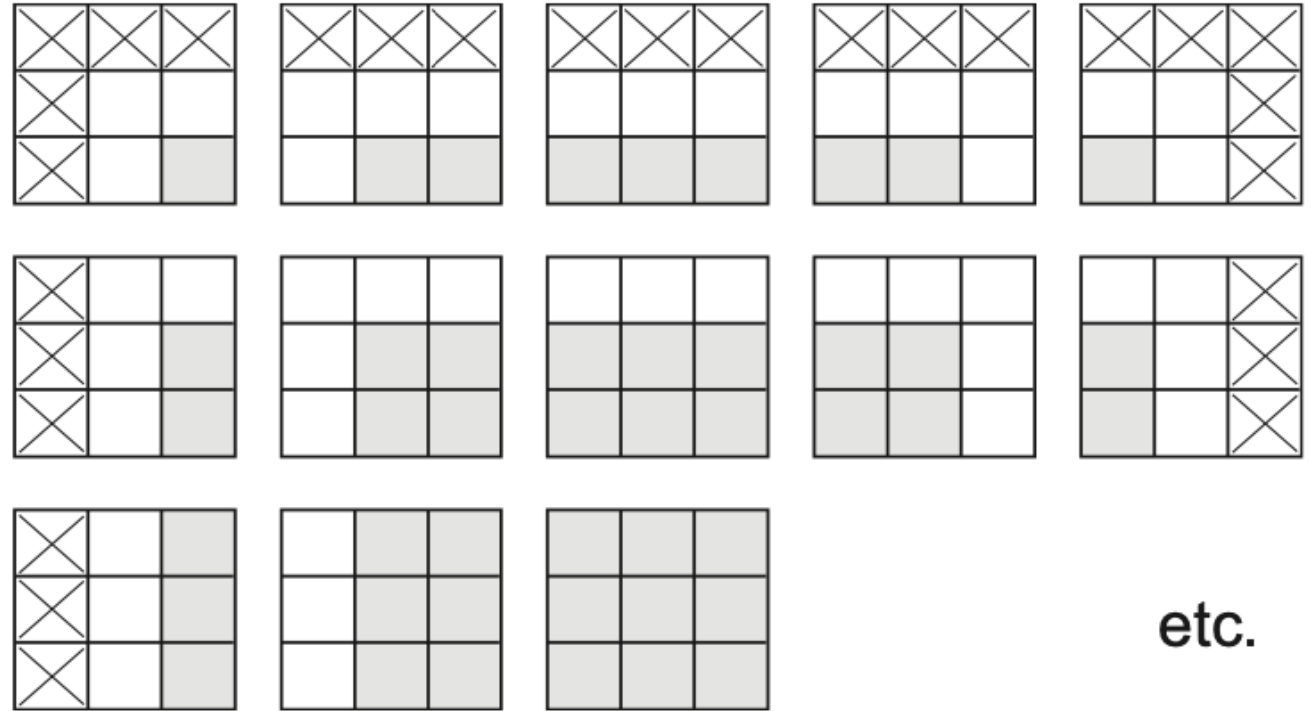
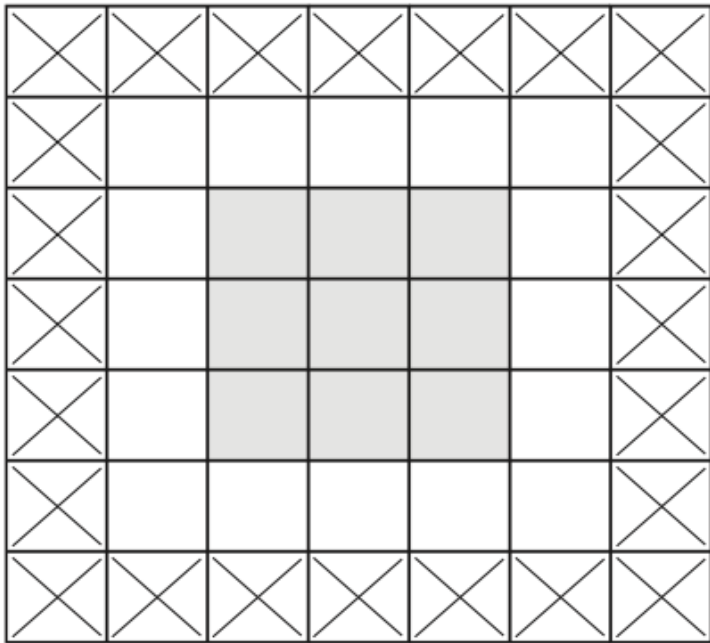
Convolution + Pooling



Stride und Max-Pooling

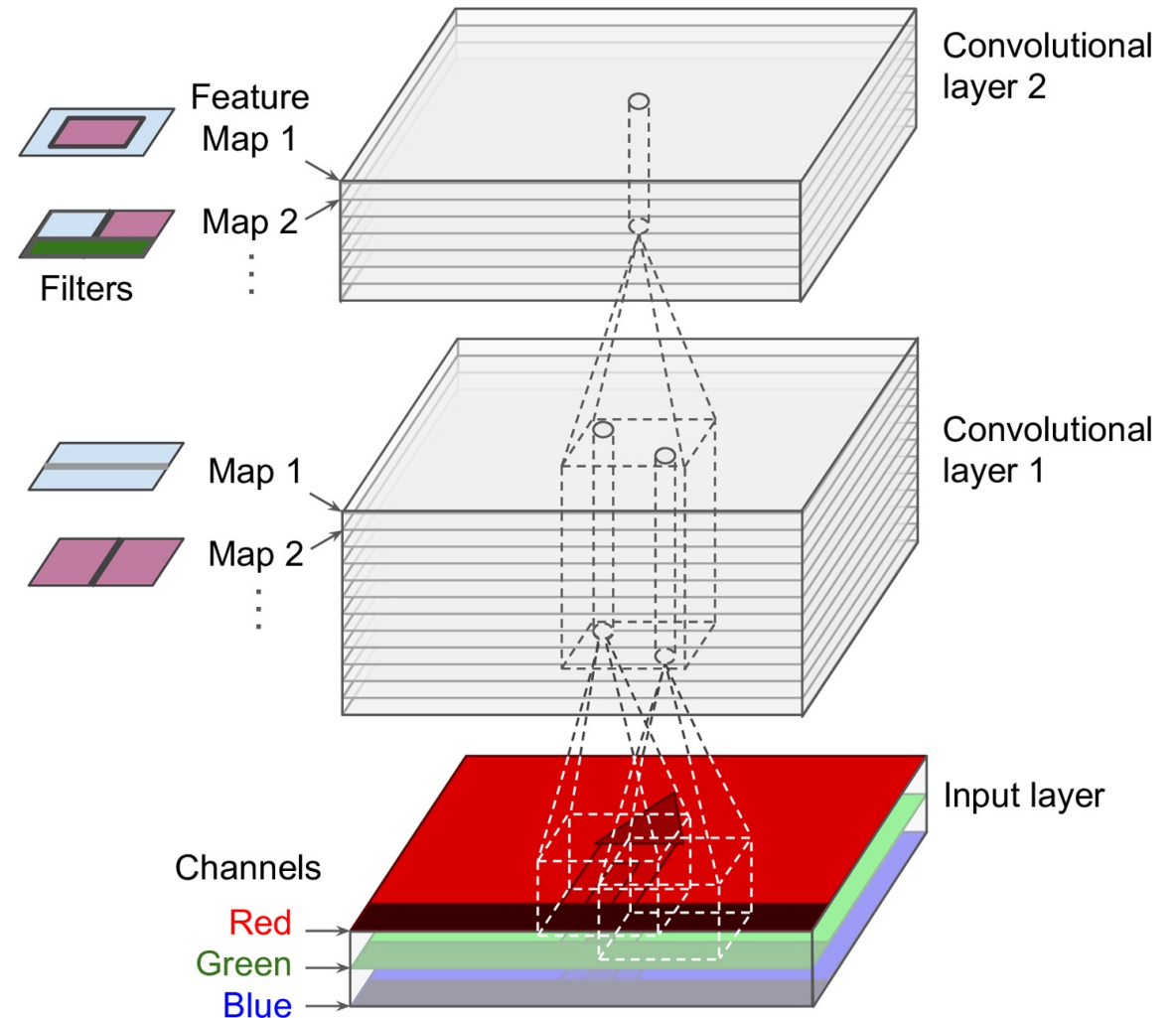


Stride, Max-Pooling und Padding

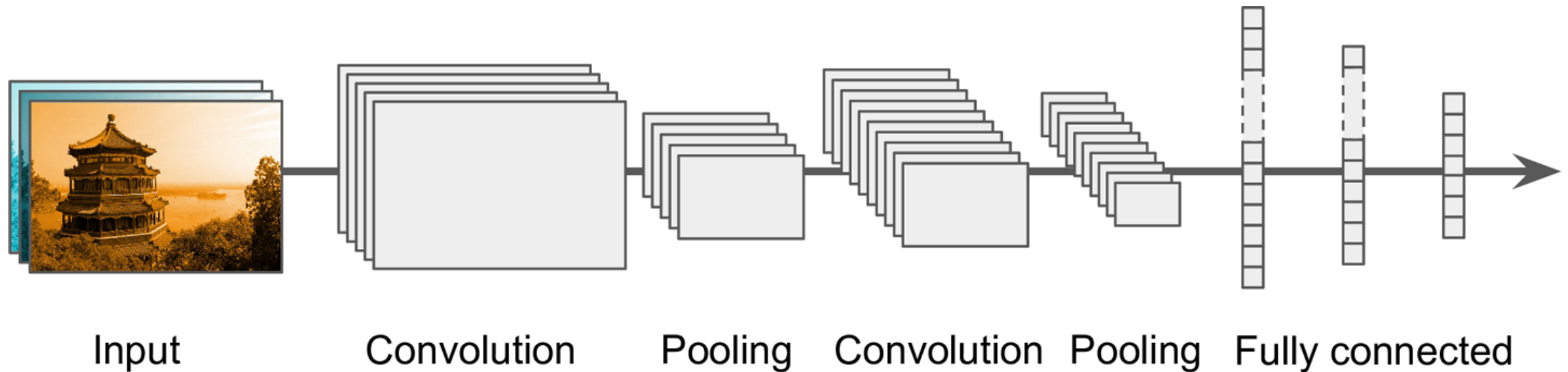


Convolutional Networks: Big Picture

- Jeder kernel/filter erzeugt eine feature map
- Outputs bilden Inputs der nächsten Layer



Convolutional Netzwerk zur Klassifikation



Convolutional Nets in der Praxis

- Bilddaten einlesen und reskalieren
- Selber trainieren
- Features verwenden
- Fine tuning