

Algorithmen für Bildanalyse: Eisbär oder Pinguin?





Grundlagen der Künstlichen Intelligenz

Wintersemester 25/26

Vorlesung 9

Convolutional Neural Networks

Anwendung: Computer Vision

Prof. Dr. Pascal Friederich
T.T.-Prof. Dr. Peer Nowack

KI-Landkarte

Künstliche Intelligenz

Modellierung und Schlussfolgerung

Variablen VL12 Inferenz

Logik Wissensrepräsentation

VL11

Zustände Suche VL13 MDPs

Reflex

Anwendungen

Robotik VL14

Computer Vision VL9 Natürliche Sprache VL10

Lernen

Optimierung und Generalisierung VL6

Vorhersage VL4 VL5 Neuronale Netze

Modellierung VL3 Supervised Unsupervised VL7 VL8

Historie und Philosophie

VL1

Geschichte

Personen

KI und Gesellschaft

Kritische Aspekte

Mathematik

VL2

Lineare Algebra

Statistik

Logik

Numerik

Analysis

Überblick

- 1) Bilddaten
- 2) Convolutional Neural Networks für Bildklassifizierung
- 3) Anwendungen, ResNet, Transfer Learning
- 4) Weitere Tasks: Objekterkennung, Segmentierung, ...

Teil 1

1) Bilddaten

2) Convolutional Neural Networks für Bildklassifizierung

3) Anwendungen, ResNet, Transfer Learning

4) Weitere Tasks: Objekterkennung, Segmentierung, ...

Bilddaten

Visualisierung: Was wir sehen



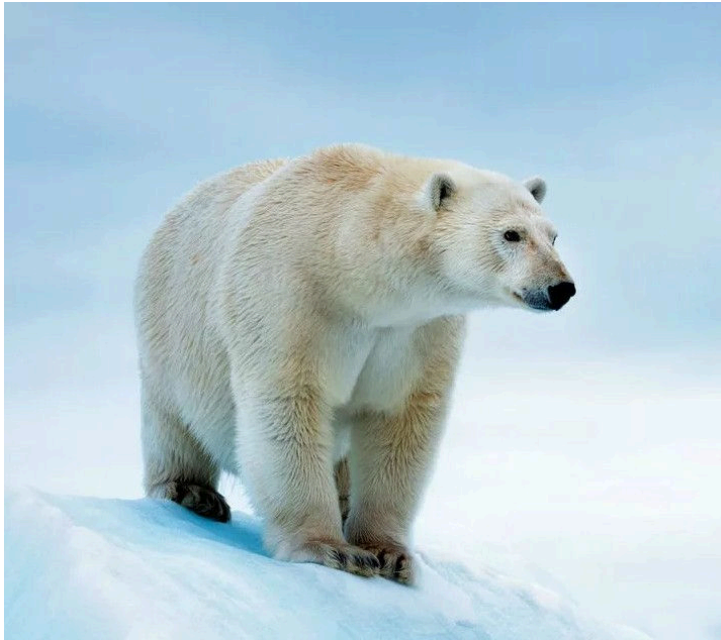
Daten: Was der Computer „sieht“

156	159	158	155	...
160	154	157	158	...
156	159	158	155	...
160	154	157	158	...
⋮	⋮	⋮	⋮	⋮

Graustufen-Werte
N x M Matrix

Bilddaten

Visualisierung: Was wir sehen



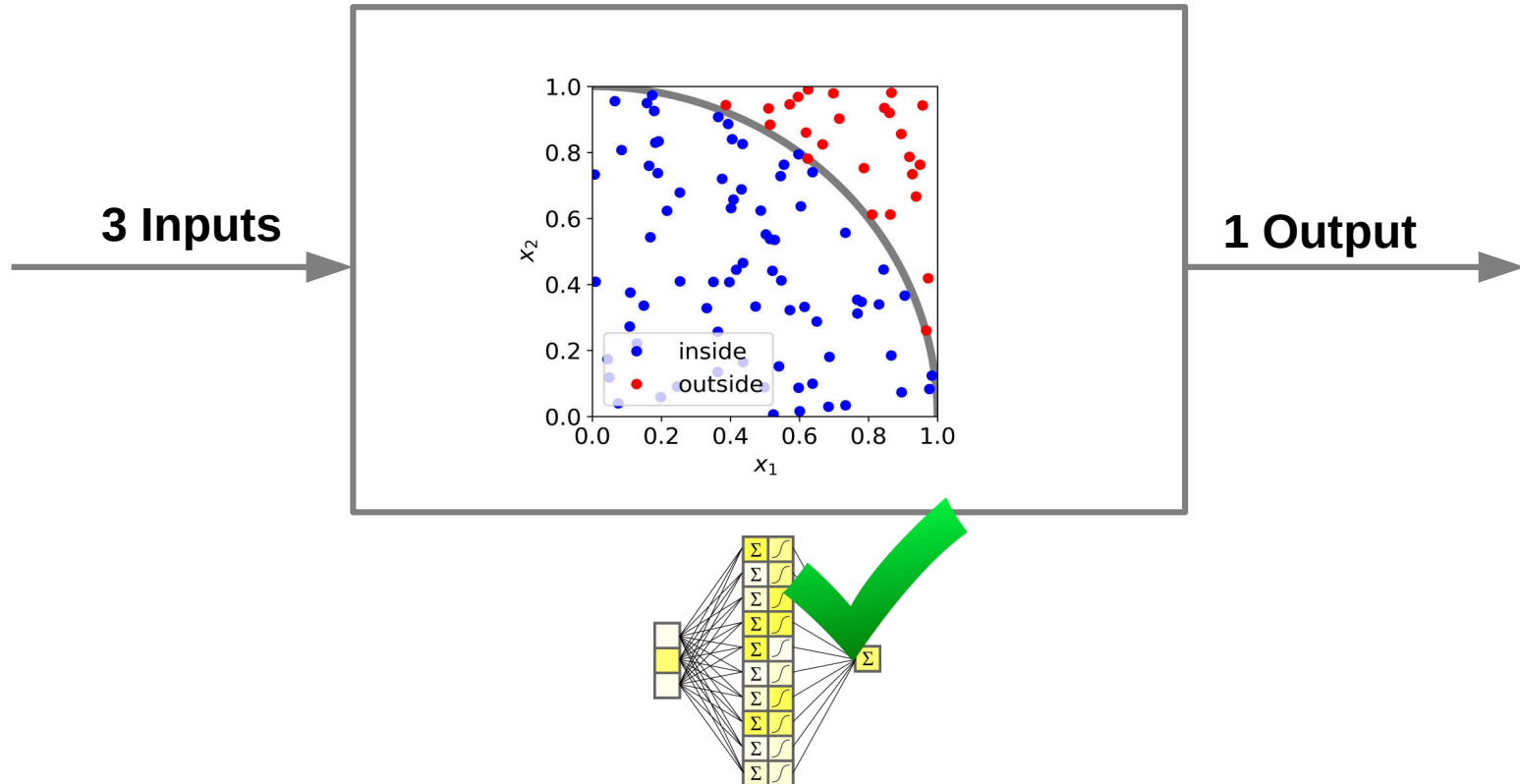
Daten: Was der Computer „sieht“

	156	159	158	155	...
	160	154	157	158	...
	156	159	158	155	...
	160	154	157	158	...
	⋮	⋮	⋮	⋮	⋮

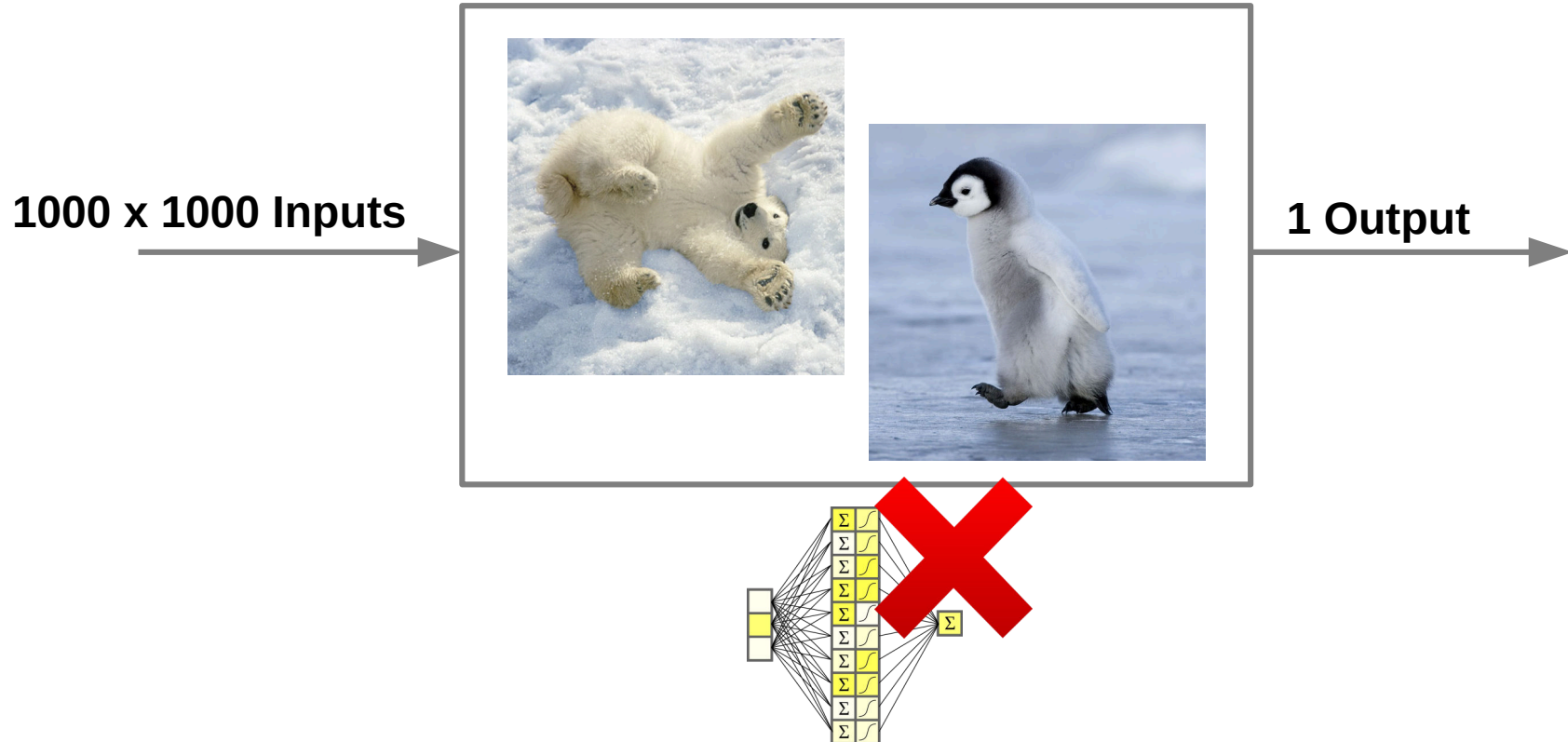
RGB-Werte (Rot, Grün, Blau)

$N \times M \times 3$ Tensor

Binäre Klassifikation für Feature Vektoren



Binäre Klassifikation für Bilder



Teil 2

1) Bilddaten

2) Convolutional Neural Networks für Bildklassifizierung

3) Anwendungen, ResNet, Transfer Learning

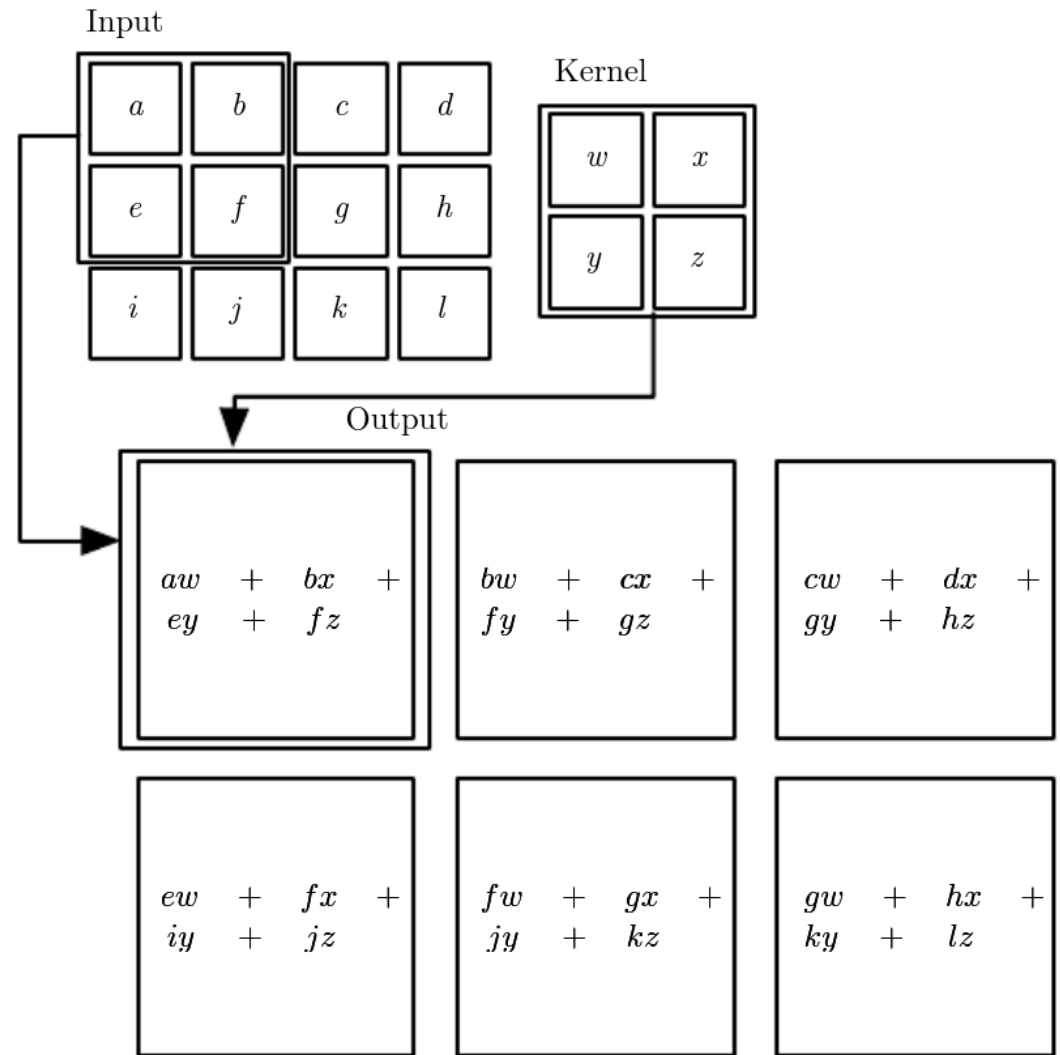
4) Weitere Tasks: Objekterkennung, Segmentierung, ...

Convolutional neural networks

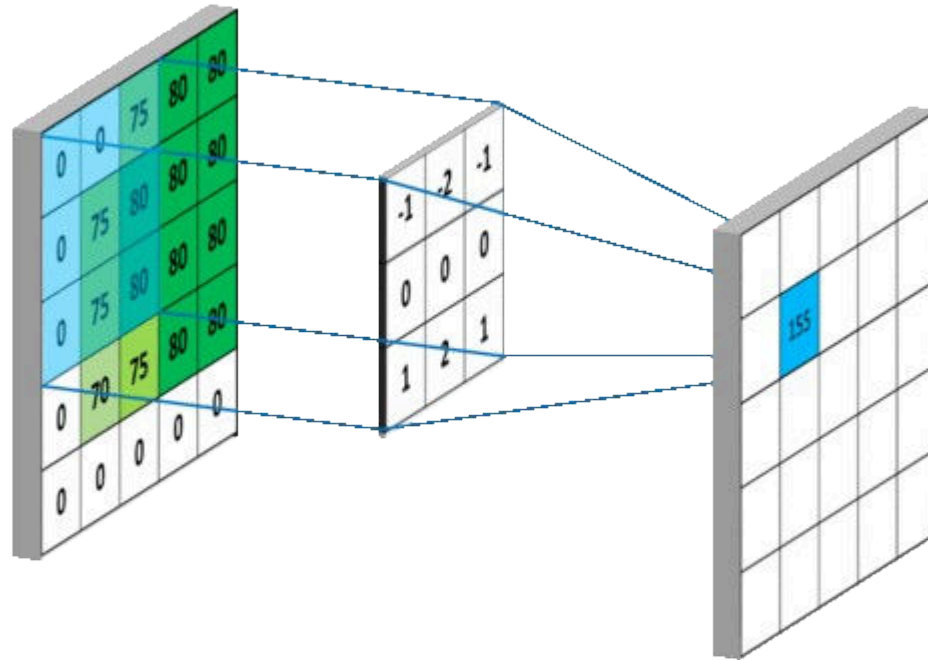
- Spezielle Architektur eines neuronalen Netzes
- Entwickelt für Daten auf Grids (1D, 2D, 3D, ...)
- Abgekürzt: **CNNs**

Illustration

- Die Eingabe (hier 3 x 4) wird mit dem Kernel/Filter (hier 2 x 2) abgetastet
- Dies erzeugt eine Ausgabe (hier 2 x 3)
- Jeder Eintrag in der Ausgabe ist die Summe der elementweisen Produkte von Eingabe und Kernel



Convolutional neural networks



Copyright: Aayushmaan Jain, medium.com

Convolutional neural networks: Faltende neuronale Netze

- Der Name bezieht sich auf die Art und Weise, wie Informationen über das Netz verbreitet werden
- Convolution / Faltung

$$\begin{array}{c} \text{Output} \\ \downarrow \\ S(t) = (I * K)(t) = \sum_a I(a)K(t - a) \end{array} \quad \begin{array}{c} \text{Input} \\ \downarrow \\ \end{array} \quad \begin{array}{c} \text{Kernel / Filter} \\ \downarrow \\ \end{array}$$

Convolutional neural networks: Faltende neuronale Netze

2D-Fall

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

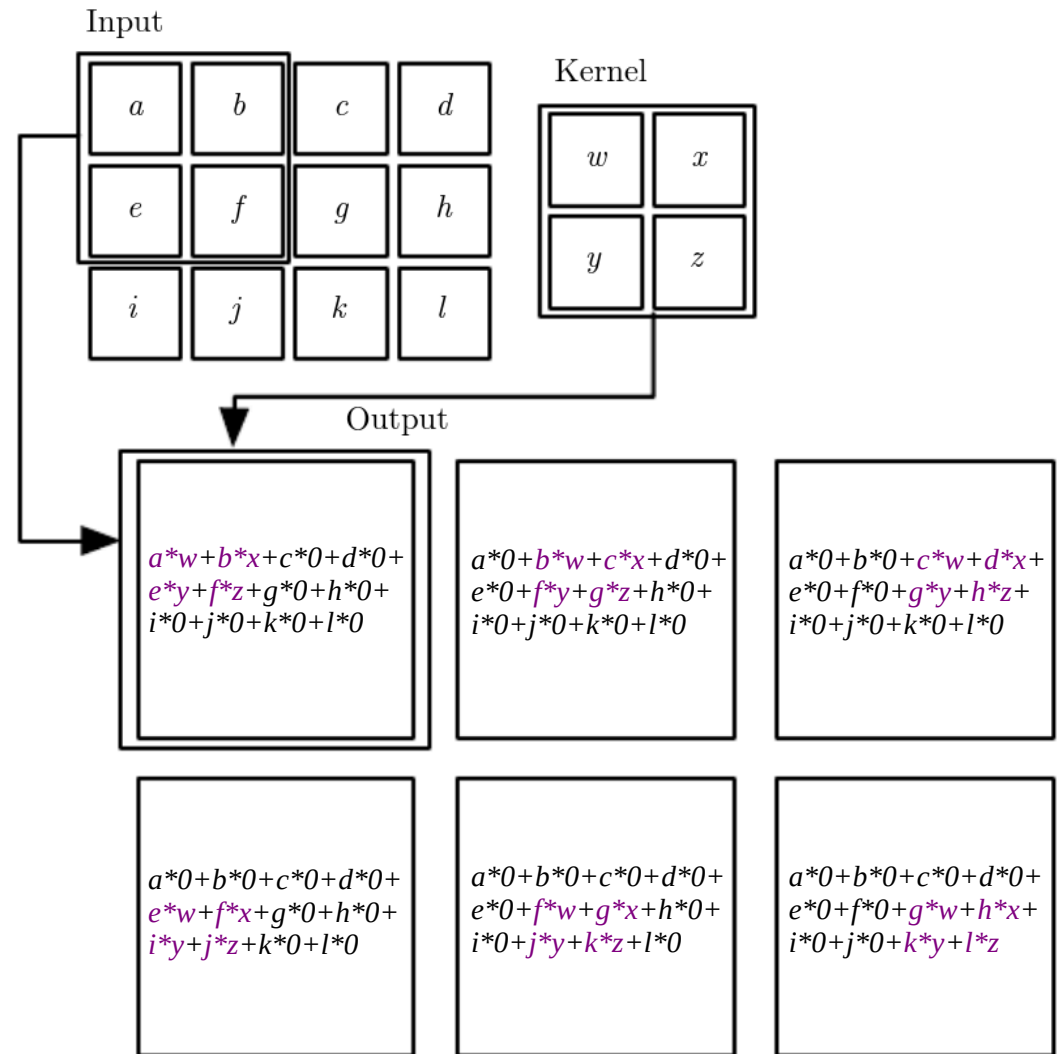
CNNs verwenden jedoch stattdessen häufig die so genannte "**cross-correlation**" (**Kreuzkorrelation**)

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Convolutional neural networks

Alternative Ansicht

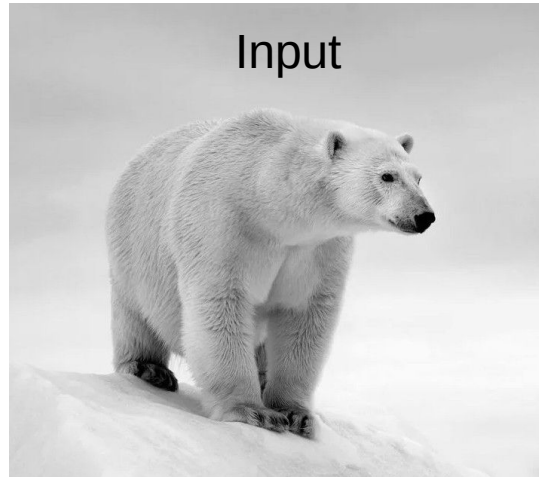
- Die funktionelle Abbildung von der Eingabe zur Ausgabe kann als "normaler" densely-connected Layer betrachtet werden.
 - Die meisten Gewichte sind Null
 - Viele weitere Gewichte sind identisch
- Abbildung von 12 auf 6 Neuronen
 - Normalerweise $6 * 12 = 72$ Gewichte
 - Hier nur 4 (also 4 Kernel Parameter)
- Zwei Prinzipien
 - "Sparse connectivity"
 - "Shared weights"



Beispiel: Kantenerkennung

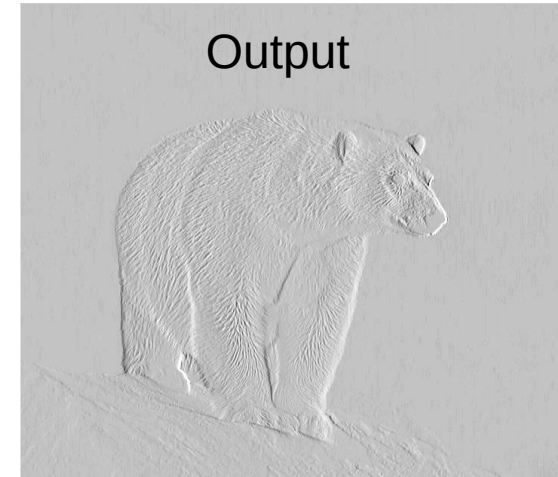
Convolutions sind sehr viel effizienter als densely-connected Layers

- Eingabe: 280 x 320
- Filter: 1 x 2 (Gewichte -1 und 1 für die Kantenerkennung)
- Ausgabe: 280 x 319
- Convolution: 2 Gewichte, $319 \cdot 280 \cdot 3 = 267.960$ Berechnungen (zwei Multiplikationen und eine Addition pro Output-Neuron)
- Dicht verbunden: $280 \cdot 320 \cdot 280 \cdot 319 = 8$ Milliarden Gewichte, 16 Milliarden Berechnungen



Filter/Kernel

-1	1
----	---



Beispiel: Kantenerkennung

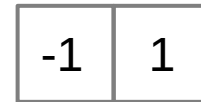
Convolutions sind sehr viel effizienter als densely-connected Layers

- Eingabe: 280 x 320
- Filter: 1 x 2 (Gewichte -1 und 1 für die Kantenerkennung)
- Ausgabe: 280 x 319
- Convolution: 2 Gewichte, $319 \cdot 280 \cdot 3 = 267.960$ Berechnungen (zwei Multiplikationen und eine Addition pro Output-Neuron)
- Dicht verbunden: $280 \cdot 320 \cdot 280 \cdot 319 = 8$ Milliarden Gewichte, 16 Milliarden Berechnungen

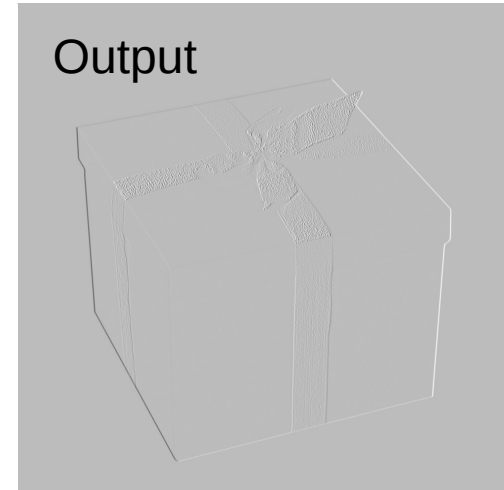
Input



Filter/Kernel



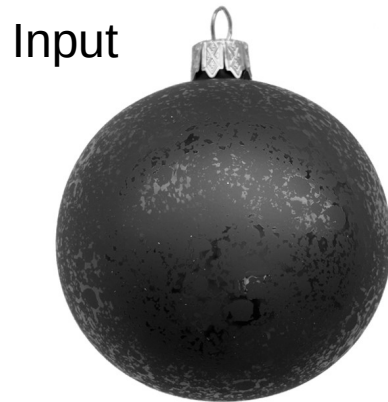
Output



Beispiel: Kantenerkennung

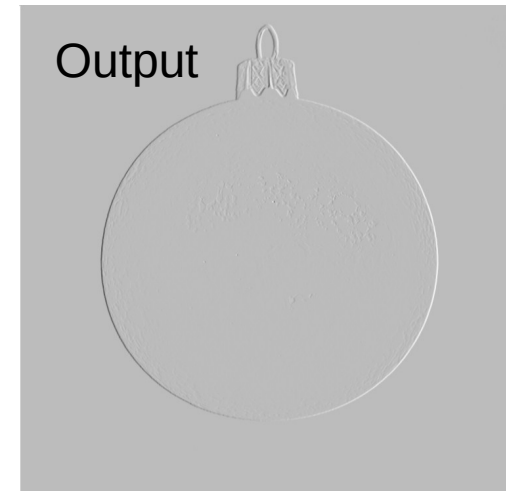
Convolutions sind sehr viel effizienter als densely-connected Layers

- Eingabe: 280 x 320
- Filter: 1 x 2 (Gewichte -1 und 1 für die Kantenerkennung)
- Ausgabe: 280 x 319
- Convolution: 2 Gewichte, $319 \cdot 280 \cdot 3 = 267.960$ Berechnungen (zwei Multiplikationen und eine Addition pro Output-Neuron)
- Dicht verbunden: $280 \cdot 320 \cdot 280 \cdot 319 = 8$ Milliarden Gewichte, 16 Milliarden Berechnungen



Filter/Kernel

-1	1
----	---



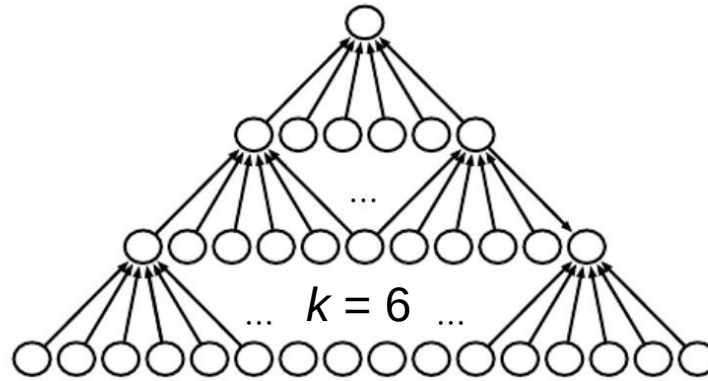
Größenreduzierung während der Faltung

$$m_4 = 1$$

$$m_3 = 6$$

$$m_2 = 11$$

$$m_1 = 16$$



Größenreduzierung während der Faltung



30 Sekunden



- Input: 200 x 400
- Filter: 5 x 5
- Output Größe?

Größenreduzierung während der Faltung



30 Sekunden



- Input: 200 x 400
- Filter: 5 x 5
- Output Größe?

196 x 396

Größenreduzierung während der Faltung



30 Sekunden



- Input: $m \times n$
- Filter: $k \times k$
- Output Größe?

Größenreduzierung während der Faltung



30 Sekunden



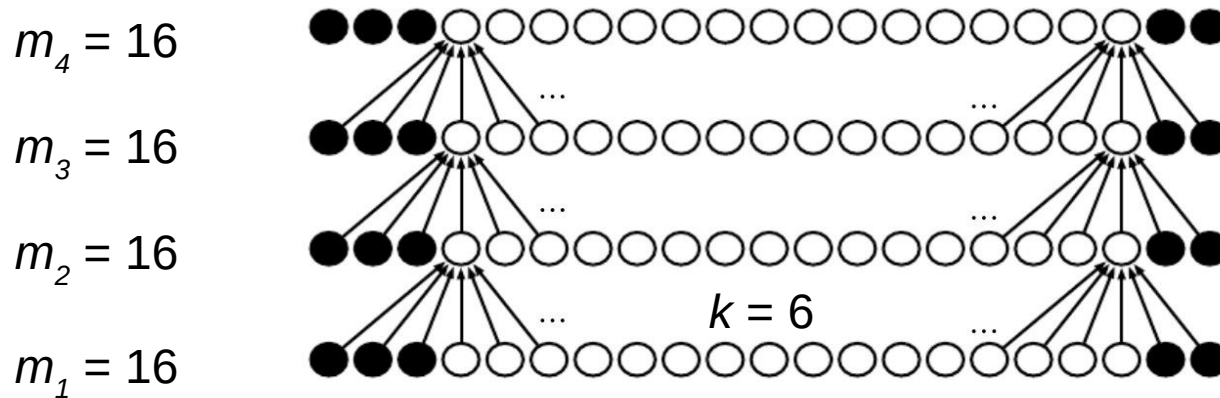
- Input: $m \times n$
- Filter: $k \times k$
- Output Größe?

$$(m - k + 1) \times (n - k + 1)$$

Größenreduzierung während der Faltung

Padding

- Erweiterung des Inputs um 0, so dass die Größe nach der Faltung gleich bleibt
- Kein padding ("**valid convolution**"): Input m , Kernel k , Output $m - k + 1$
- Zero-padding ($k - 1$ Pixel, "**same convolution**"): Größe bleibt erhalten



Pooling

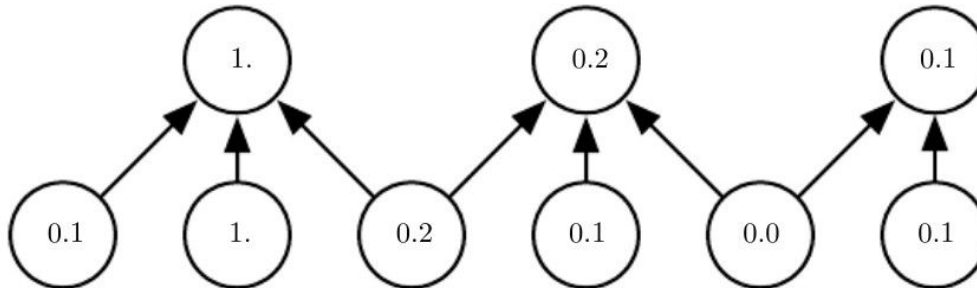
In vielen CNNs werden **Faltungen** mit **Pooling** kombiniert

→ **Kombination der Informationen mehrerer Pixel zu einem einzigen**

- Max pooling: Neuer Pixelwert ist das Maximum der vorherigen Pixel
- Average pooling: Das neue Pixelwert ist der Durchschnitt der vorherigen Pixel
- L_2 pooling: Neuer Pixelwert ist L_2 Norm der vorherigen Pixel
- ...

Typischer Zweck

- Starke Größenreduzierung ("downsampling"), mit *width* (z.B. 3) und *stride* (z.B. 2)



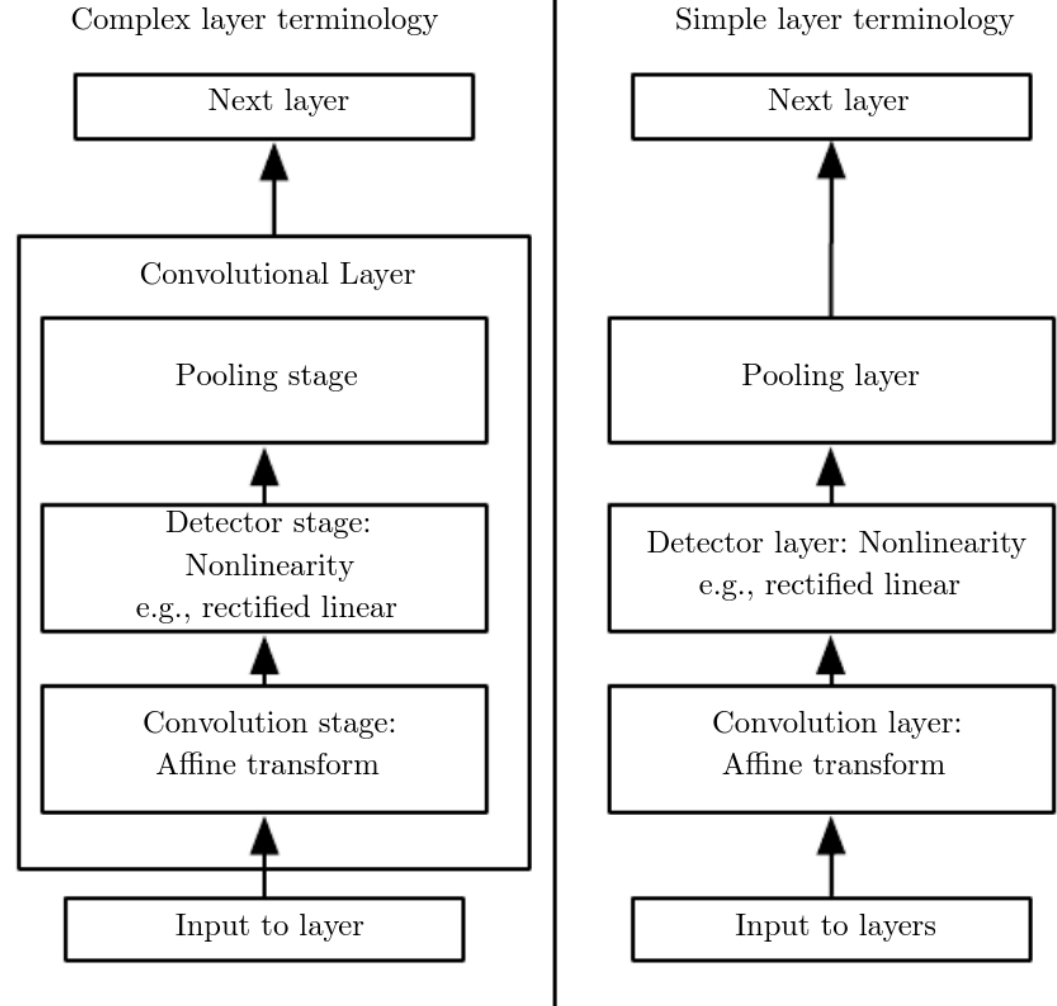
Randnotiz:

Stride kann auch bei Convolution verwendet werden!

Nomenklatur

Kombination mehrerer Operationen zu einer Schicht

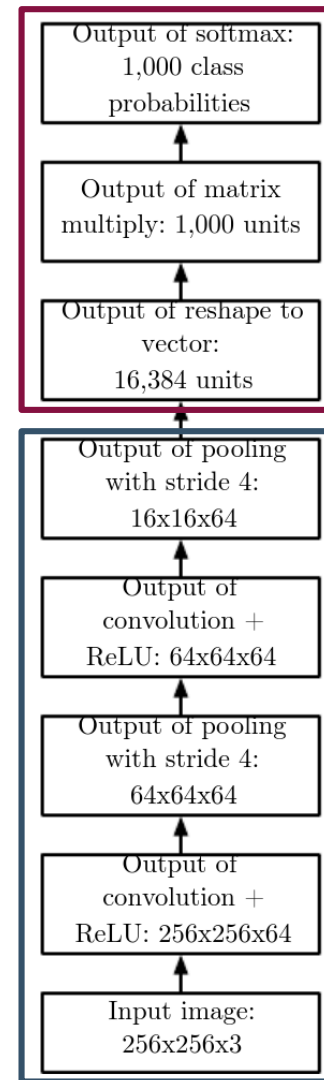
1. Convolution
 2. Nichtlinearität
 3. Pooling
- "Convolutional layer"



Überblick über gesamtes CNN model

Kombination aus

- **Convolution Layers**
 - Convolutions (mehrere Filter pro Schicht)
 - Nichtlineare Transformationen
 - Pooling
- **NN Layers**
 - Vektorisierung
 - Dense layers
 - Softmax (für Mehrklassen-Klassifizierung)



Anzahl der Gewichte in den Filtern

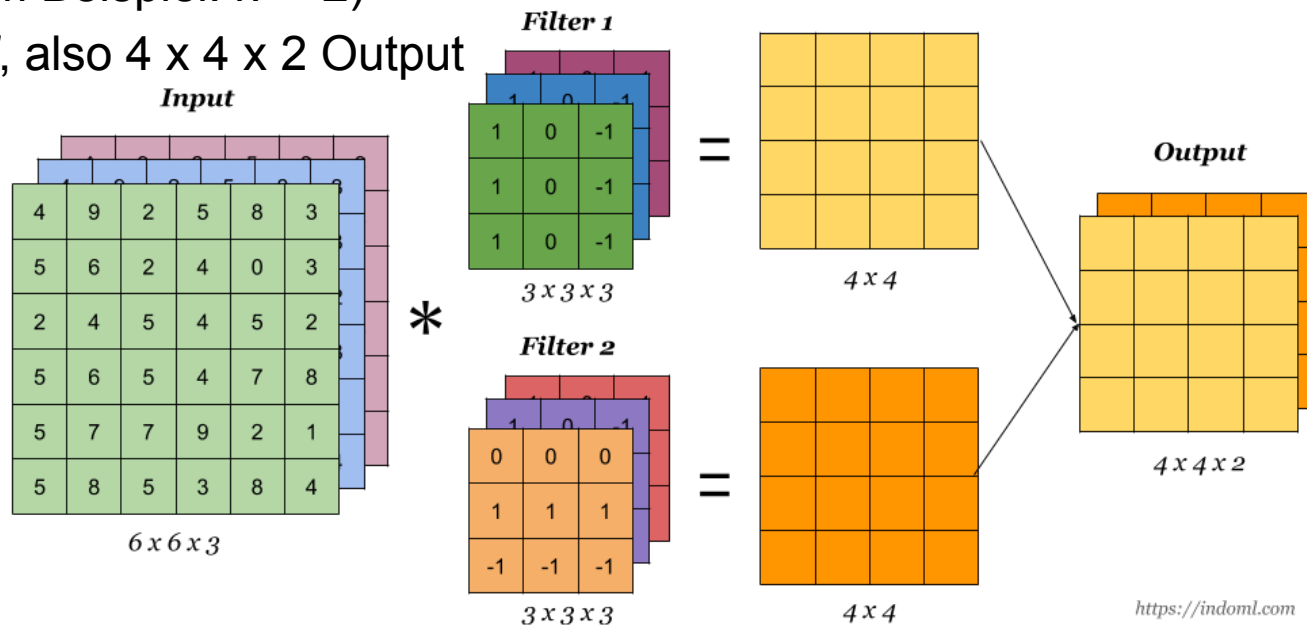
Beispiel

Bild $m \times n$ (im Beispiel: 6×6) mit k (3) (Farb)kanälen

- Input: $6 \times 6 \times 3$
- $p \times q$ Filter hat $p \times q \times 3 + 1$ **Gewichte/Parameter** (im Beispiel: $p = 3, q = 3$)
- Jeder Faltungsschritt: k' Filter (im Beispiel: $k' = 2$)
- Anzahl der Kanäle im Output: k' , also $4 \times 4 \times 2$ Output
- Insgesamt $p \times q \times k'$ Parameter

Bemerkung

- Pooling reduziert die Bildgröße während Zahl der Kanäle gleich bleibt
- $k' 1 \times 1$ Filter reduzieren die Zahl der Kanäle von k auf k' bei fixer Bildgröße



<https://indoml.com>

Konkretes Beispiel: AlexNet

ImageNet

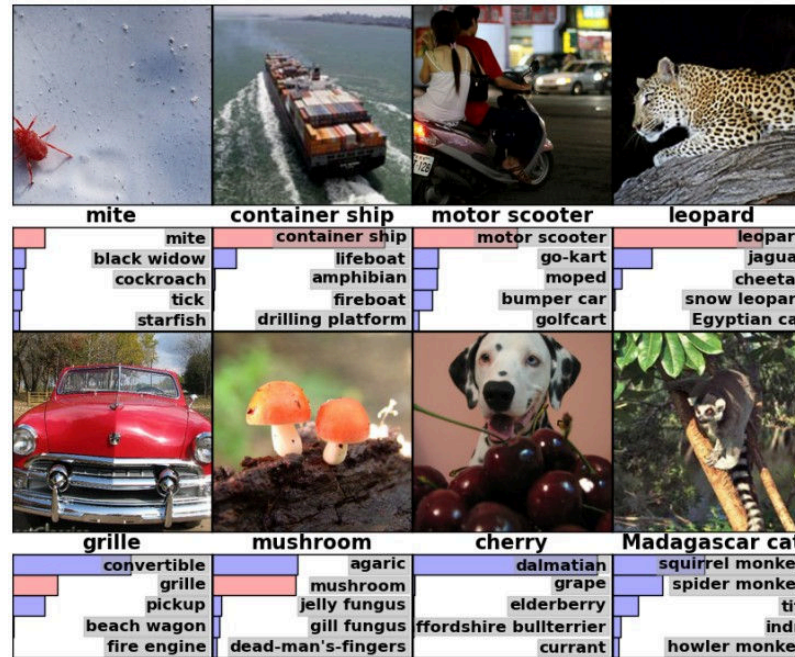
- 1,3 Millionen Bilder
- 1000 Klassen
- NeurIPS 2012
- CNN Durchbruch

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

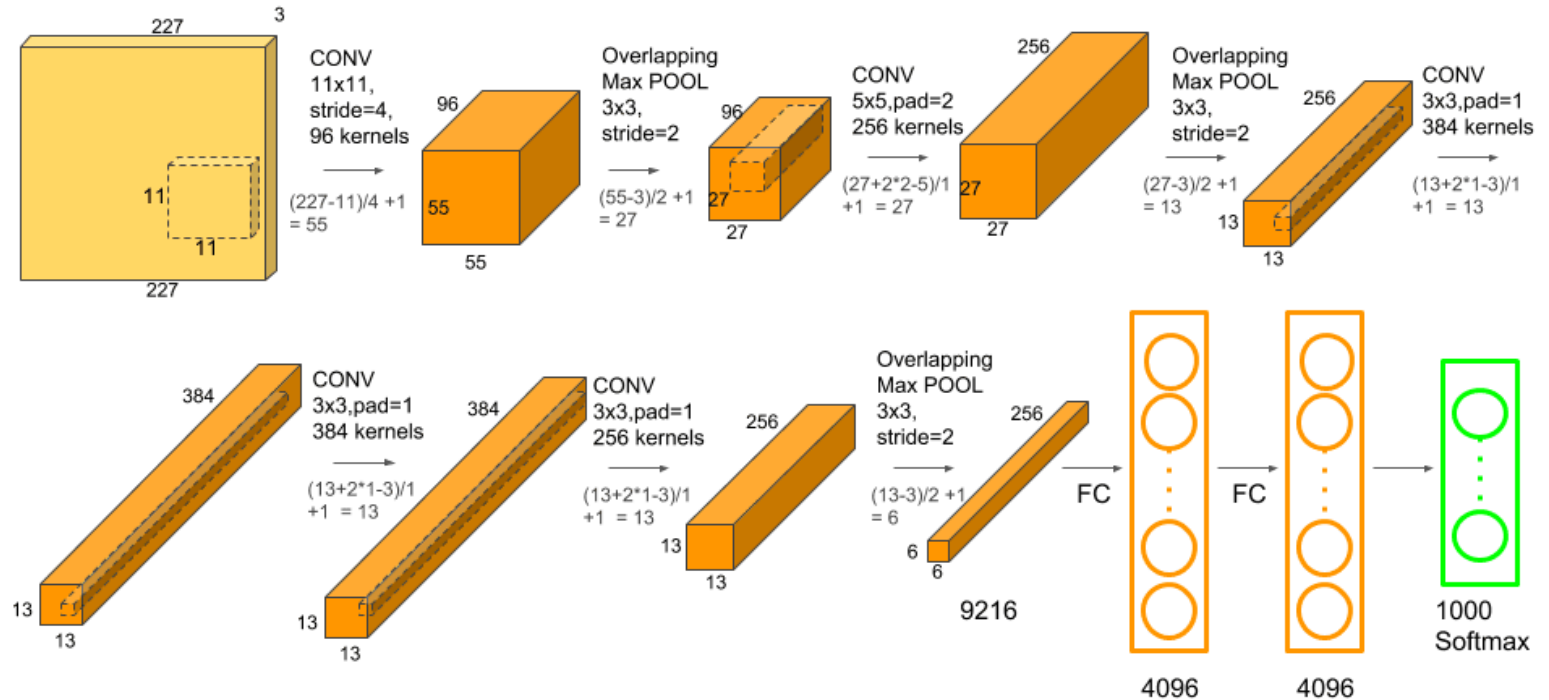
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



Konkretes Beispiel: AlexNet

ImageNet

- 1,3 Millionen Bilder
- 1000 Klassen
- NeurIPS 2012
- CNN Durchbruch

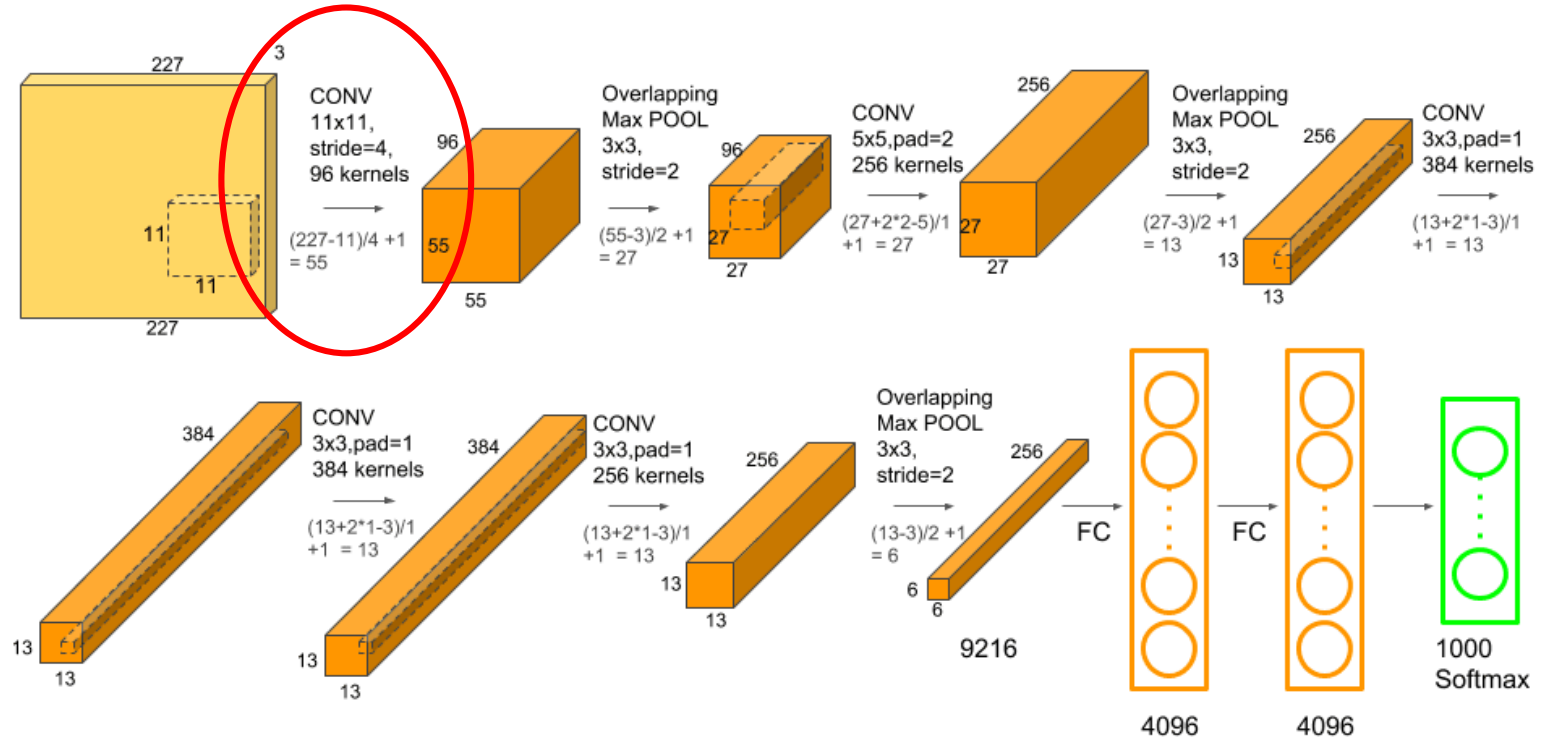


Konkretes Beispiel: AlexNet

Wie viele Gewichte sind in Layer 1? (ohne Bias)



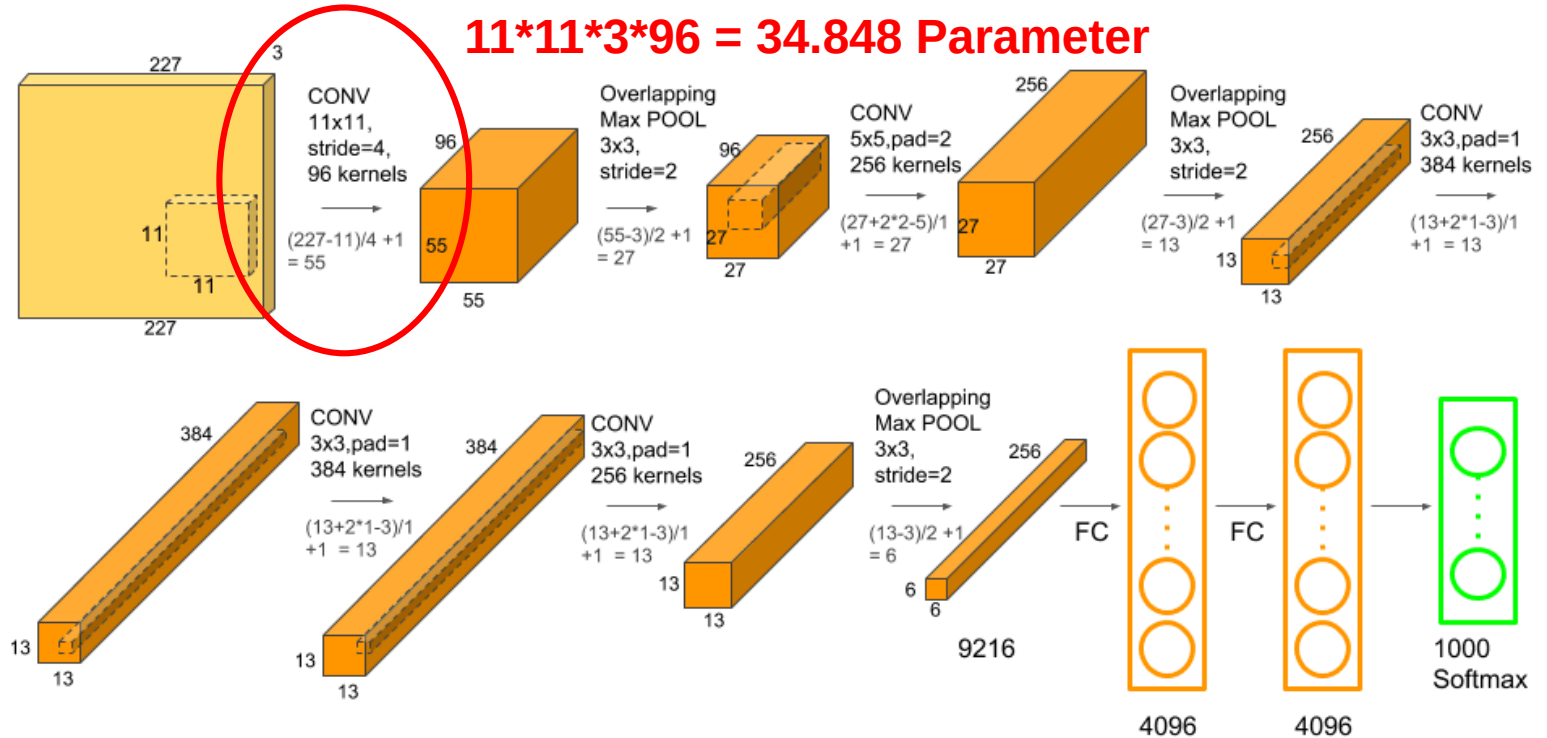
30 Sekunden



Konkretes Beispiel: AlexNet

Wie viele Gewichte sind in Layer 1? (ohne Bias)

$$11 \cdot 11 \cdot 3 \cdot 96 = 34.848 \text{ Parameter}$$



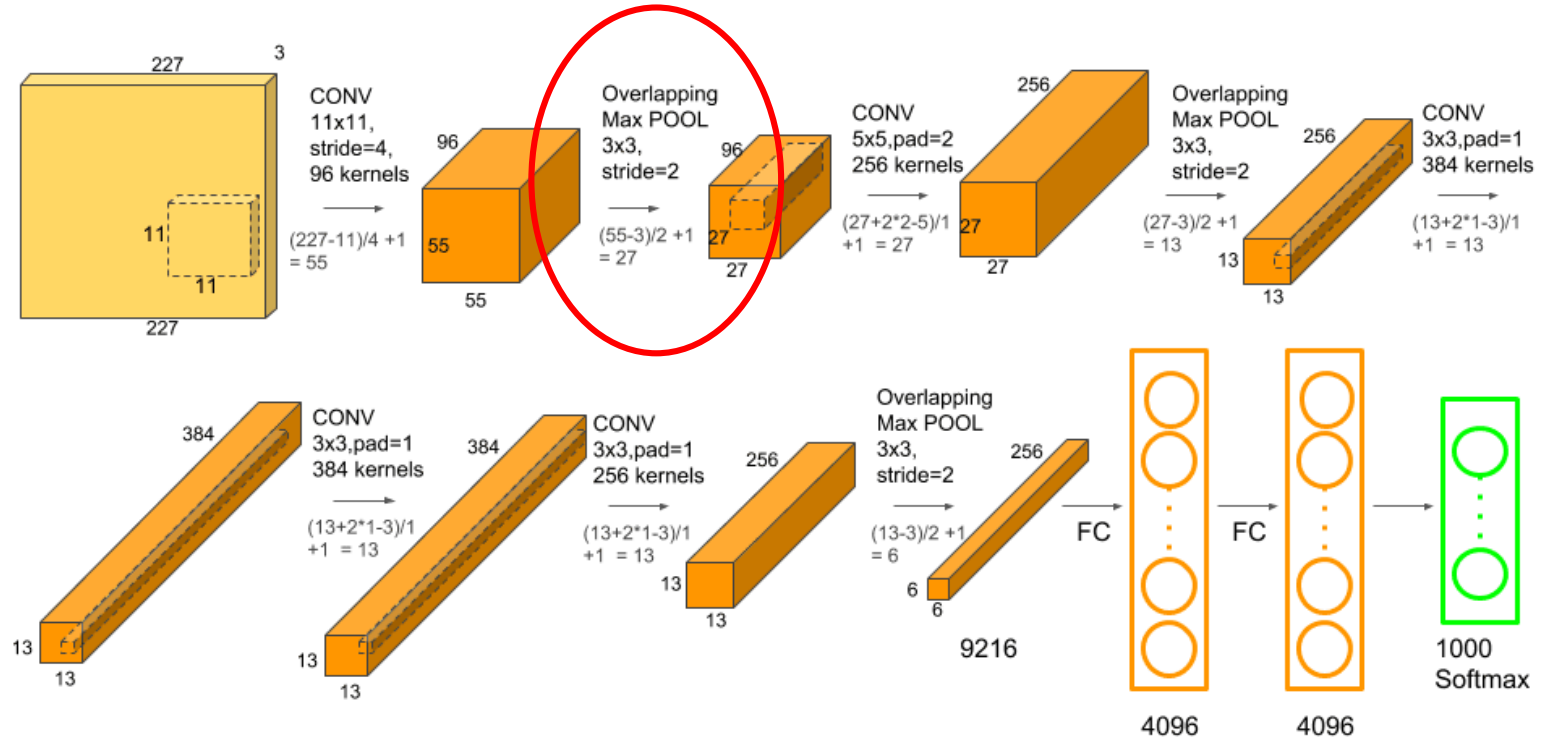
30 Sekunden

Konkretes Beispiel: AlexNet

Wie viele Gewichte sind in Layer 2? (ohne Bias)



30 Sekunden



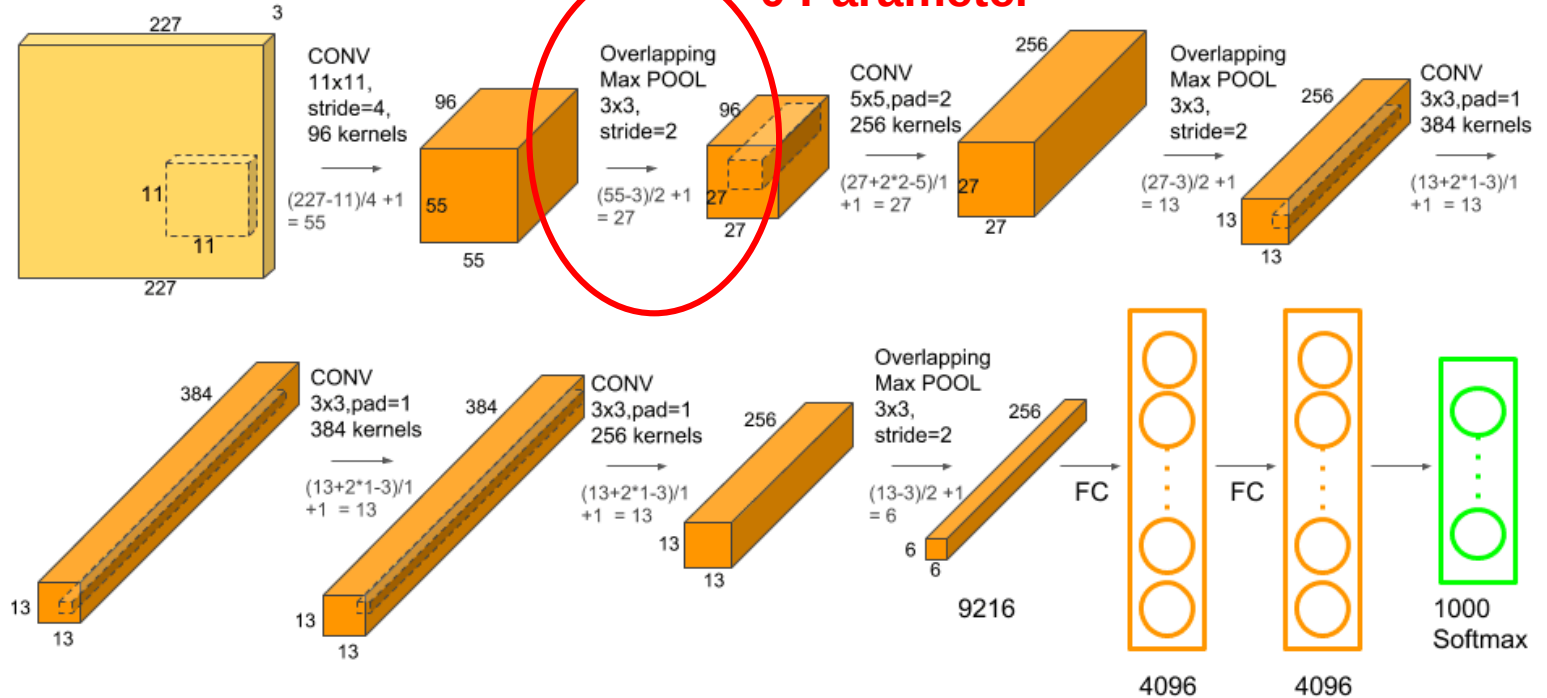
Konkretes Beispiel: AlexNet

Wie viele Gewichte sind in Layer 2? (ohne Bias)

0 Parameter

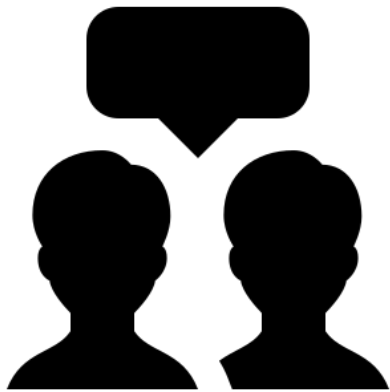


30 Sekunden

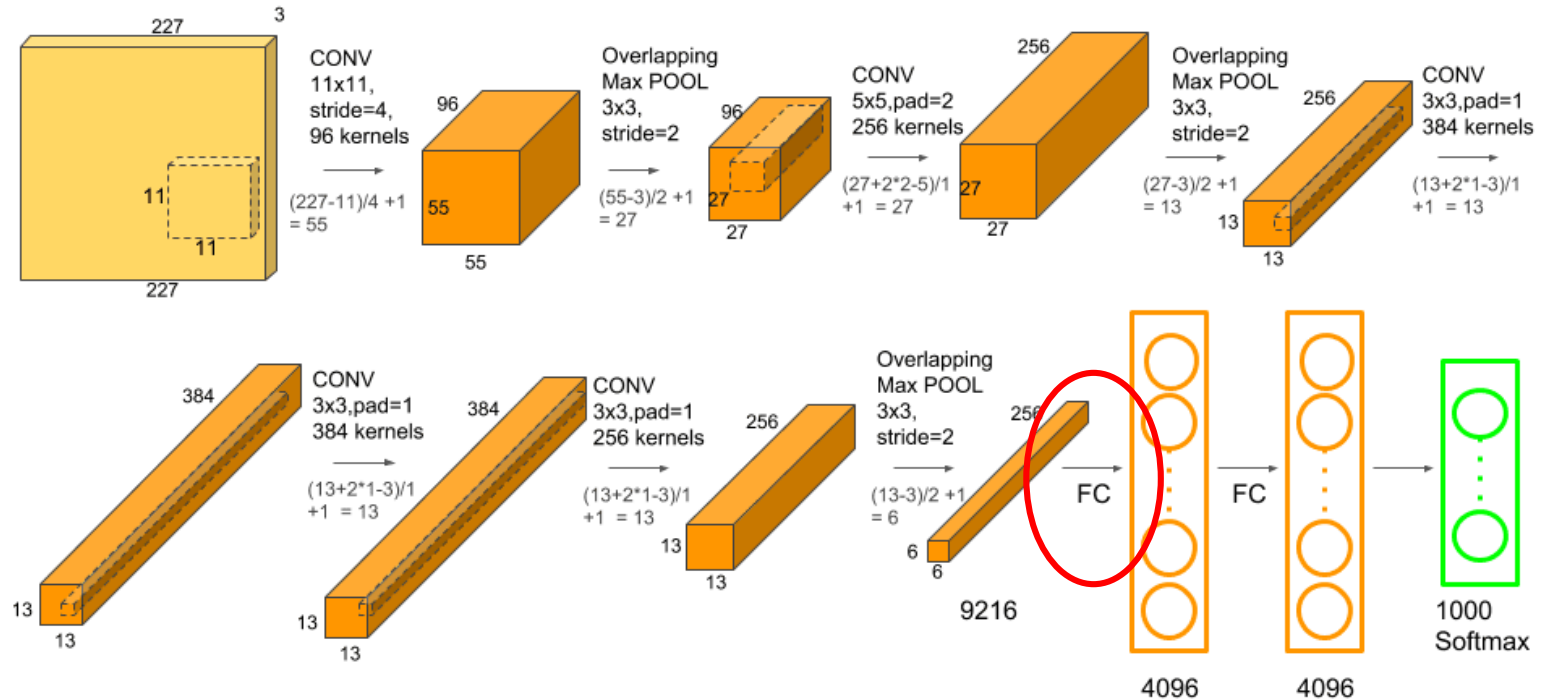


Konkretes Beispiel: AlexNet

Wie viele Gewichte sind in Layer FC1? (ohne Bias)



30 Sekunden

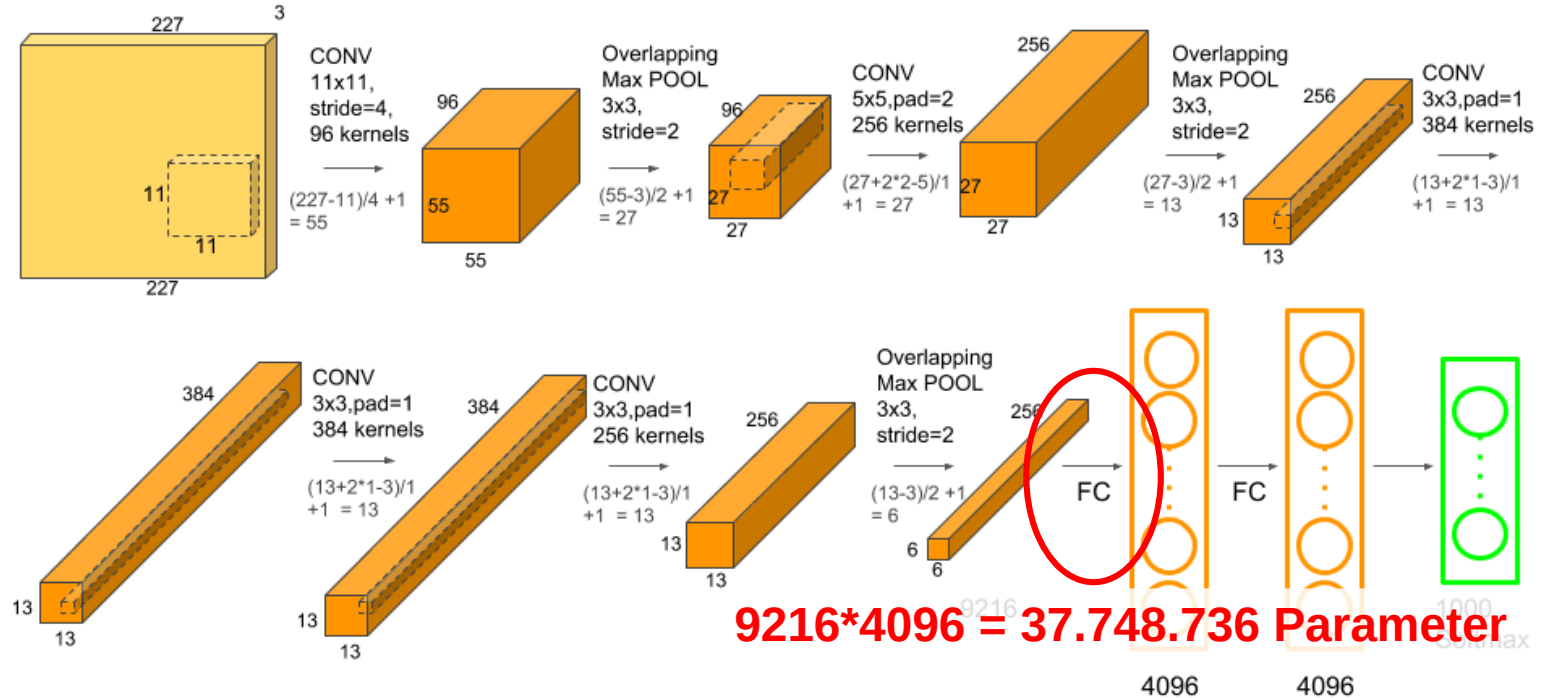


Konkretes Beispiel: AlexNet

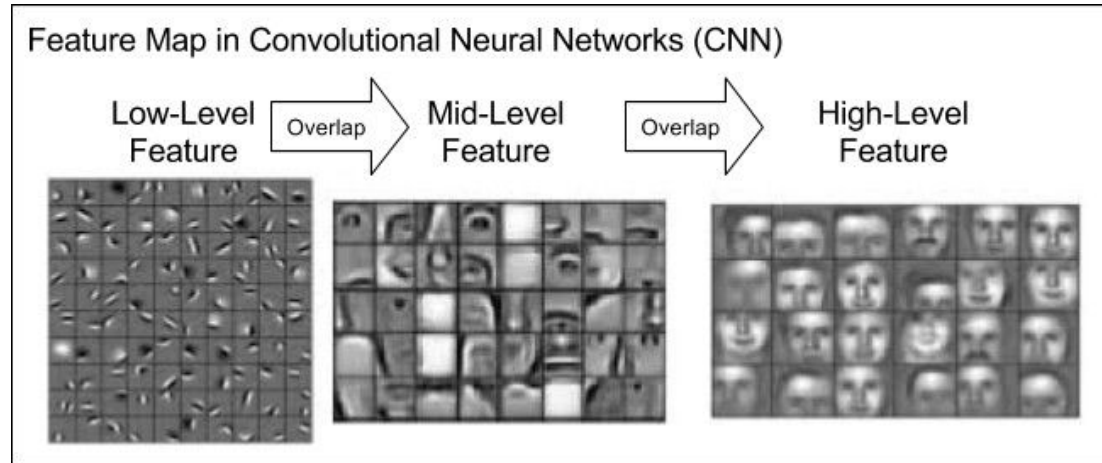
Wie viele Gewichte sind in Layer FC1? (ohne Bias)



30 Sekunden



Welche Filter werden tatsächlich gelernt?



DOI: 10.1109/ICEngTechnol.2017.8308186

Datentypen

1D

- Zeitreihen, z. B. Audiodaten
- Spektraldaten

2D

- Bilddaten, Graustufen ($l \times b$)

2D + Kanäle

- Farbige Bilder ($l \times b \times 3$), dritte Dimension: Farbwerte (Zahl der Kanäle)

3D oder höher

- Volumetrische 3D-Daten ($l \times b \times h$)
- Zwei Raumrichtungen und eine Zeitrichtung, also Videodaten (auch hier sind mehrere Farbkanäle möglich, $l \times b \times t \times 3$)
- Andere Kombinationen von Dimensionen

Teil 3

- 1) Bilddaten
- 2) Convolutional Neural Networks für Bildklassifizierung
- 3) Anwendungen, ResNet, Transfer Learning**
- 4) Weitere Tasks: Objekterkennung, Segmentierung, ...

Bildanalyse in der Medizin

CNNs: Wichtige Rolle im medizinischen Bereich

- **Bildgebende Verfahren**

- Röntgenbild
- Ultraschall
- Computertomographie (CT)
- Magnetresonanztomographie (MRT)
- Optische Mikroskopie

- **Art der Bilddaten**

- Oberflächenbilder (Haut, ...)
- Projektionen (Röntgenbilder)
- Querschnitte
- 3D-Tomographie
- Bewegte Bilder (Mikroskopie, Ultraschall, ...)

13,400
Ergebnisse zu
"cancer CNNs"
(2023: >72,000)



The screenshot shows a search engine interface with the query 'cancer cnns' in the search bar. Below the search bar, it indicates 'About 13.400 results (0,06 sec)'. The results list several academic papers, each with a title, authors, and a brief abstract. An arrow points from the text '13,400 Ergebnisse zu "cancer CNNs" (2023: >72,000)' to the search results area.

Search query: cancer cnns

About 13.400 results (0,06 sec)

Automatic detection of early esophageal cancer with CNNs using transfer learning
S Van Riel, F Van Der Sommen... - 2018 25th IEEE ... , 2018 - [ieeexplore.ieee.org](#)
The incidence of Esophageal Adenocarcinoma (EAC), a form of esophageal cancer, has rapidly increased in recent years. Dysplastic tissue can be removed endoscopically at an early stage, and since survival chances of patients are limited at later stages of the disease ...
☆ ⓘ Cited by 10 Related articles All 3 versions

3D G-CNNs for pulmonary nodule detection
M Winkels, TS Cohen - arXiv preprint arXiv:1804.04656, 2018 - [arxiv.org](#)
... These 3D G-CNNs were applied to the problem of false positive reduction for pulmonary nodule ... dataset size tested, the G-CNN achieved a FROC score close to the CNN trained on ... Lung cancer is currently the leading cause of cancer-related death worldwide, accounting for an ...
☆ ⓘ Cited by 34 Related articles All 4 versions ⓘ

Discriminative localization in CNNs for weakly-supervised segmentation of pulmonary nodules
X Feng, J Yang, AF Laine, ED Angelini - International conference on ..., 2017 - Springer
... that locate and segment nodules of various sizes can assist radiologists in cancer malignancy diagnosis ... It was very recently discovered in a study [5] on natural images that CNNs trained on ... the up-sampled weighted activation maps from the last convolutional layer in a CNN ...
☆ ⓘ Cited by 54 Related articles All 10 versions

Fast capsnet for lung cancer screening
A Mobiny, H Van Nguyen - ... Conference on Medical Image Computing and ..., 2018 - Springer
... to less than 5% for late-stage disease [1]. A major challenge in lung cancer screening is ... Our work shows that CapsNet is a promising alternative to CNN. Experimental results demonstrate that CapsNets compare favorably to CNNs when the training size is large, but significantly ...
☆ ⓘ Cited by 57 Related articles All 5 versions

Detection and recognition for life state of cell cancer using two-stage cascade CNNs
H Hu, Q Guan, S Chen, Z Ji... - IEEE/ACM transactions on ..., 2017 - [ieeexplore.ieee.org](#)
Cancer cell detection and its stages recognition of life cycle are an important step to analyze cellular dynamics in the automation of cell based-experiments. In this work, a two-stage hierarchical method is proposed to detect and recognize different life stages of bladder cells ...
☆ ⓘ Cited by 5 Related articles All 5 versions

Beispiel: Große CNN Studie mit Daten aus den USA und GB

- 2020 in Nature veröffentlicht
- Durchgeführt von Google Health, DeepMind, Imperial College London, Northwestern, Cambridge, Stanford, ...

Article

International evaluation of an AI system for breast cancer screening

<https://doi.org/10.1038/s41586-019-1799-6>

Received: 27 July 2019

Accepted: 5 November 2019

Published online: 1 January 2020



Scott Mayer McKinney^{1,14*}, Marcin Sieniek^{1,14}, Varun Godbole^{1,14}, Jonathan Godwin^{2,14}, Natasha Antropova², Hutan Ashrafian^{3,4}, Trevor Back², Mary Chesus², Greg C. Corrado¹, Ara Darzi^{3,4,5}, Mozziyar Etemadi⁶, Florencia Garcia-Vicente⁶, Fiona J. Gilbert⁷, Mark Halling-Brown⁸, Demis Hassabis², Sunny Jansen⁹, Alan Karthikesalingam¹⁰, Christopher J. Kelly¹⁰, Dominic King¹⁰, Joseph R. Ledsam², David Melnick⁶, Hormuz Mostofi¹, Lily Peng¹, Joshua Jay Reicher¹¹, Bernardino Romera-Paredes², Richard Sidebottom^{12,13}, Mustafa Suleyman², Daniel Tse^{1*}, Kenneth C. Young⁸, Jeffrey De Fauw^{2,15} & Shravya Shetty^{1,15*}

Verfahren und Daten

Daten aus den USA und GB

- 1) Vergleich mit Testdaten
- 2) GB → USA

Test datasets

		
Number of women	25,856	3,097
Interpretation	Double reading	Single reading
Screening interval	3 years	1 or 2 years
Cancer follow-up	39 months	27 months
Number of cancers	414 (1.6%)	686 (22.2%)

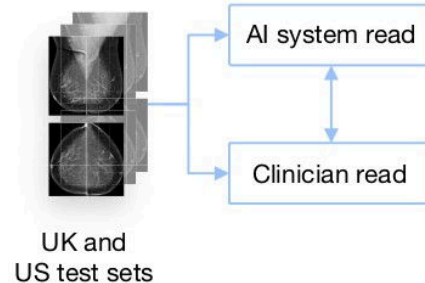
Ground-truth determination



Evaluation

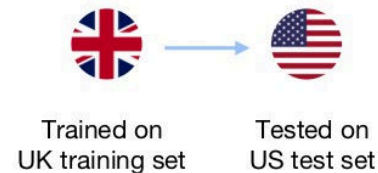
1)

Comparison with retrospective clinical performance



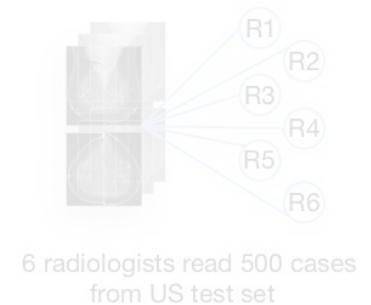
2)

Generalization across datasets



3)

Independently conducted reader study



Ergebnisse

ROC-Kurven im Vergleich zu ExpertInnen

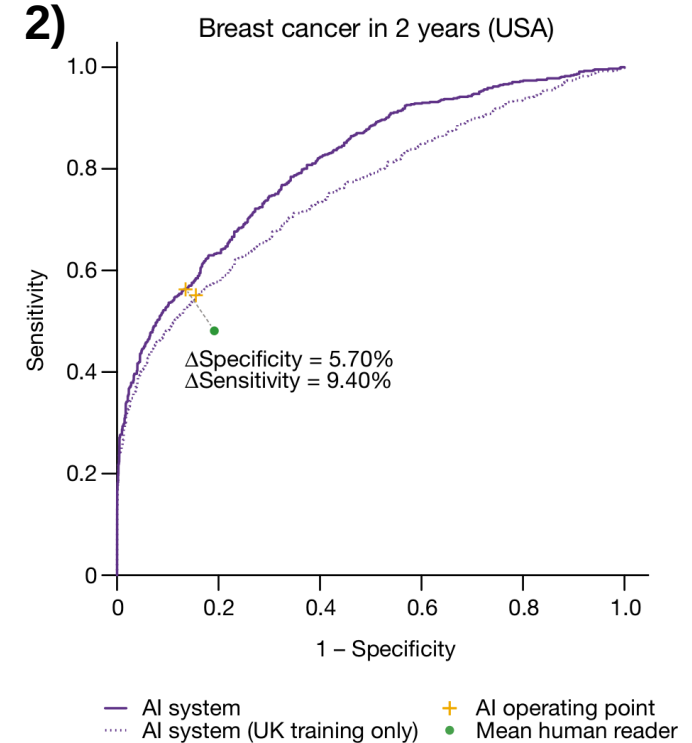
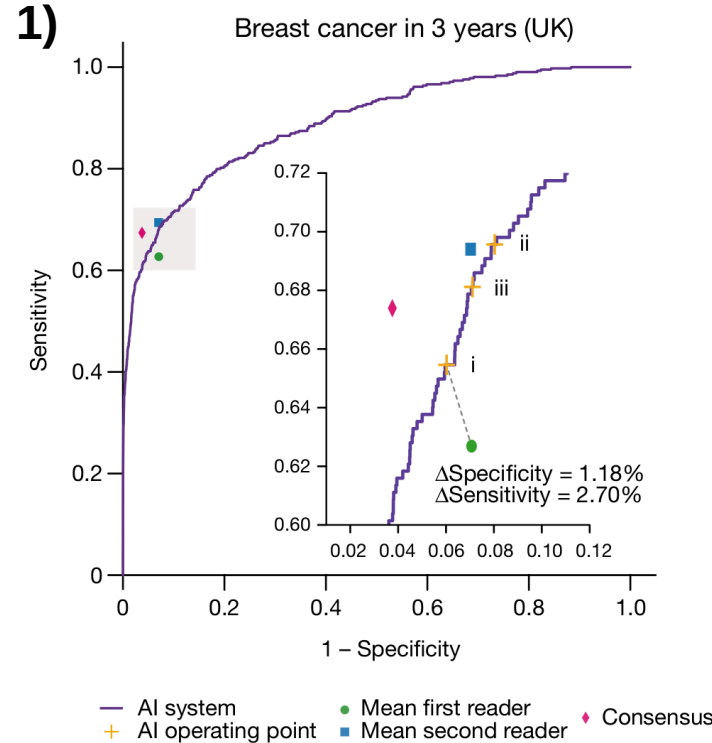
- True positive rate *über*
- False positive rate

GB

- Erste Analyse (grün)
- Zweite Analyse (blau)
- Konsens durch dritte Analyse (rot)

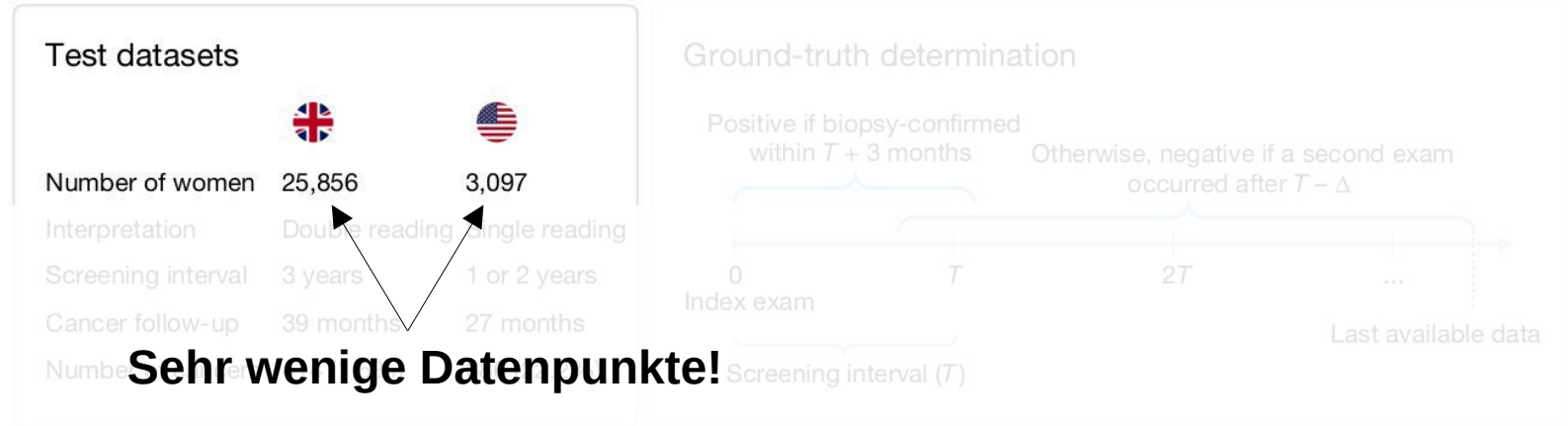
USA

- Nur eine Analyse (grün)



CNNs sind genauer als ein Experte/eine Expertin, aber schlechter als zwei oder mehr

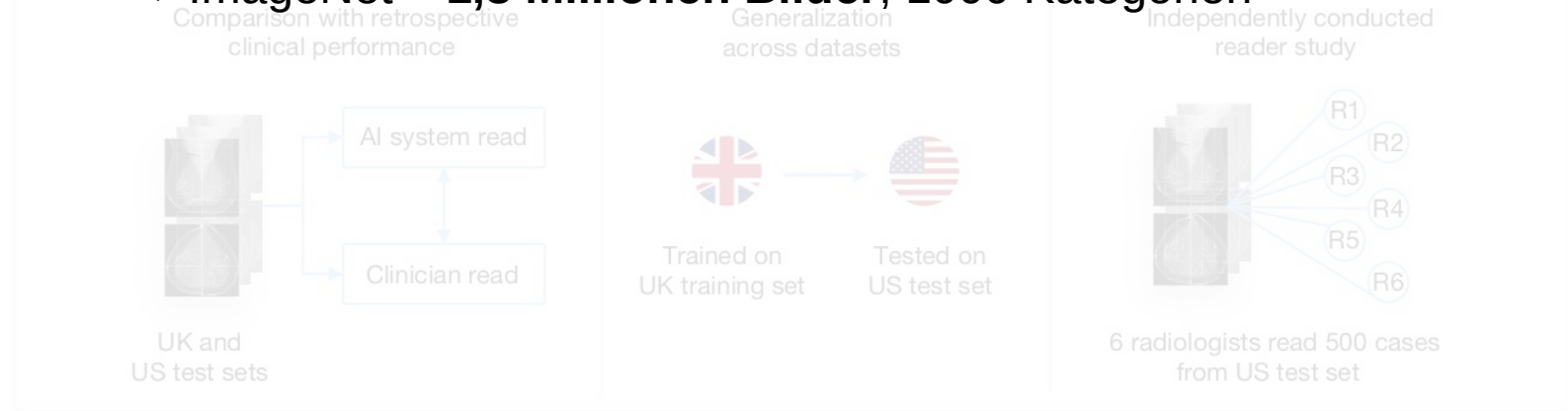
Warum sind die CNNs so gut?



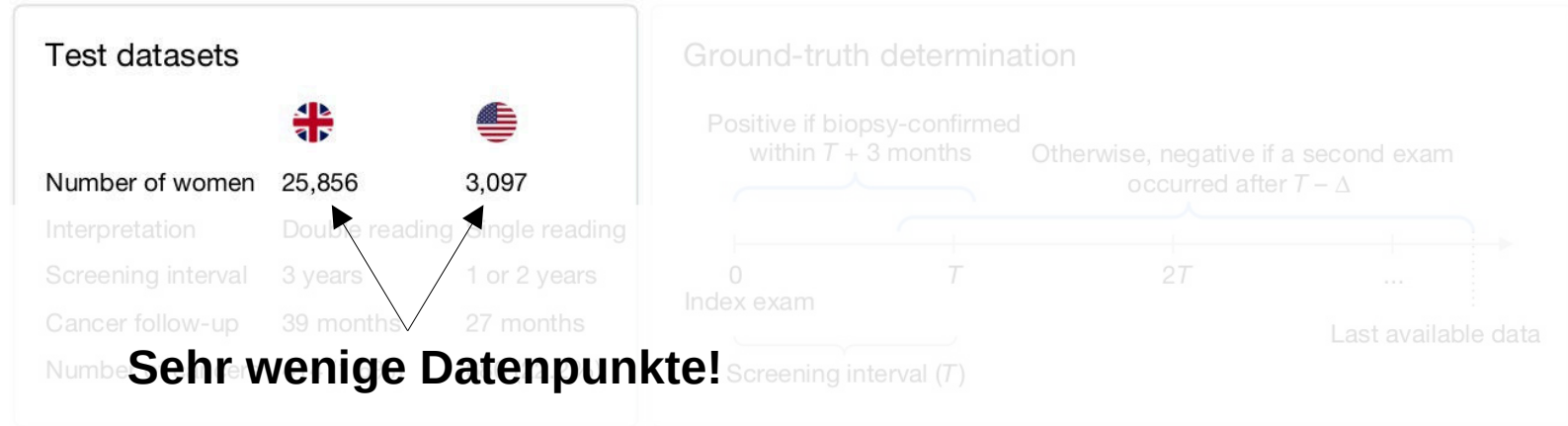
Vergleich

Evaluation

→ ImageNet – 1,3 Millionen Bilder, 1000 Kategorien



Warum sind die CNNs so gut?



Vergleich

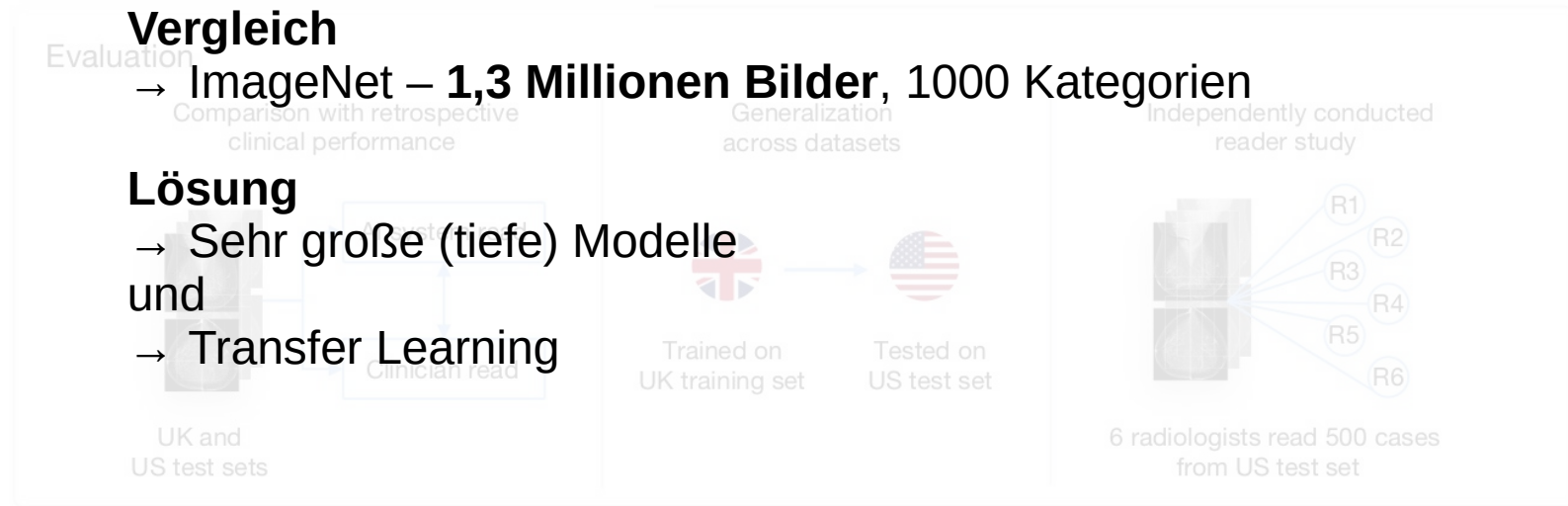
→ ImageNet – 1,3 Millionen Bilder, 1000 Kategorien

Lösung

→ Sehr große (tiefe) Modelle

und

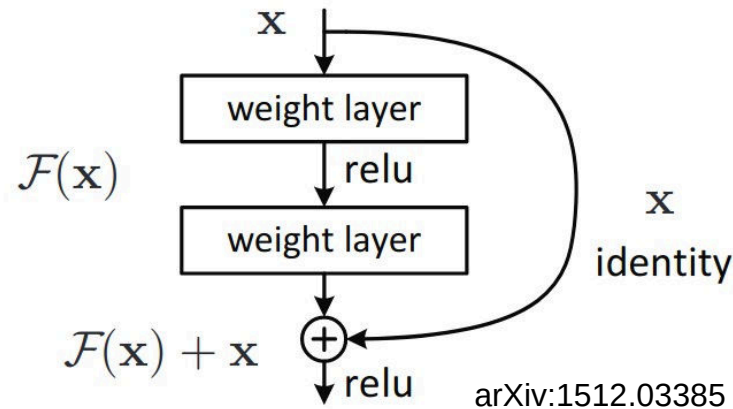
→ Transfer Learning



Residual learning: ResNet

Idee

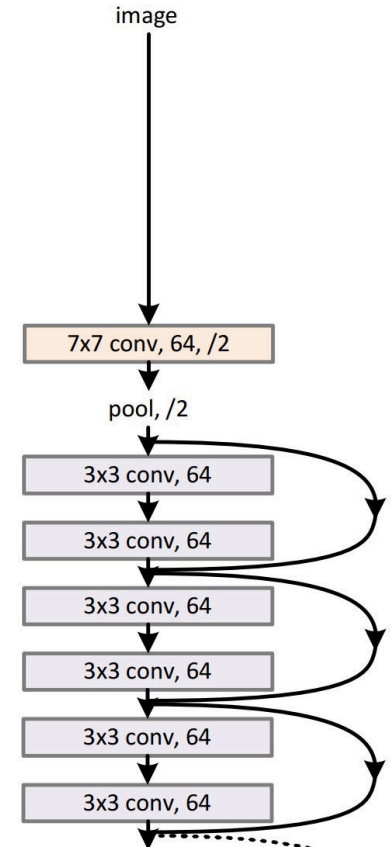
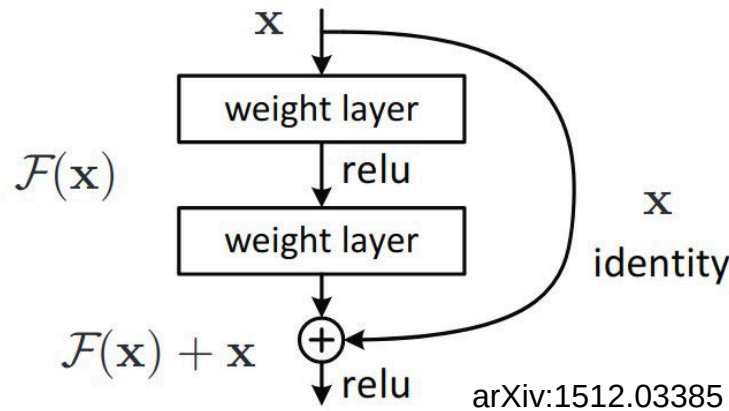
- Korrektur der Eingabe lernen (anstatt kompletter Transformation)
- Layer: Input x , Output $x + F(x)$
- $F(x)$ ist ein CNN Block (ohne Pooling)



Residual learning: ResNet

Begründung

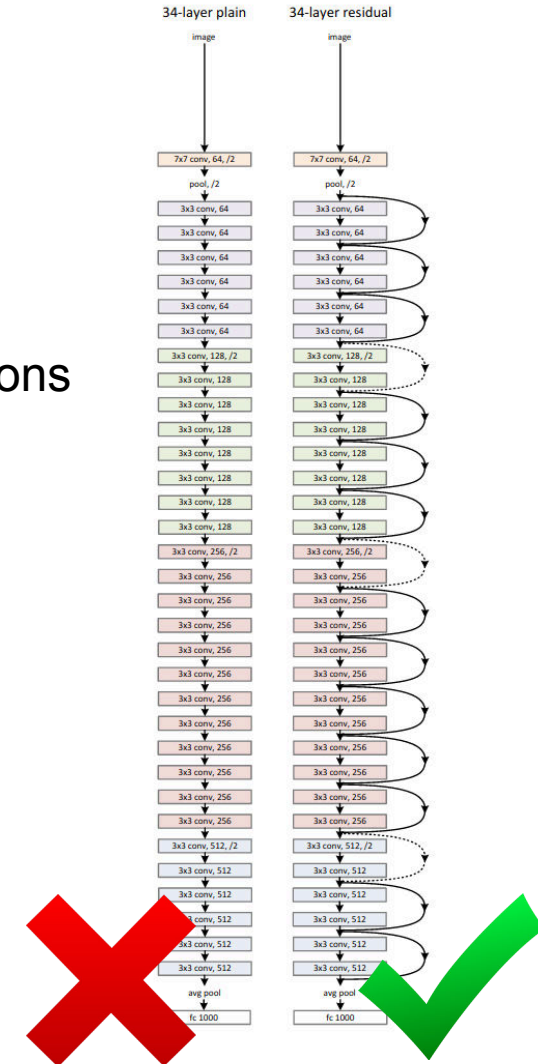
- Informationen werden weitergegeben
- Gelernte Änderungen werden addiert
- Backpropagation: Direkter Pfad von Loss zu jeder einzelnen Schicht
- Jede Schicht wird effizient trainiert → Schnelles Training trotz Tiefe



Residual learning: ResNet

Damit

- Residual connections für extrem tiefe neuronale Netze
- → CNNs mit bis zu 1000 trainierbaren Schichten
- Nicht trainierbar ohne residual connections/skip connections



Transfer Learning

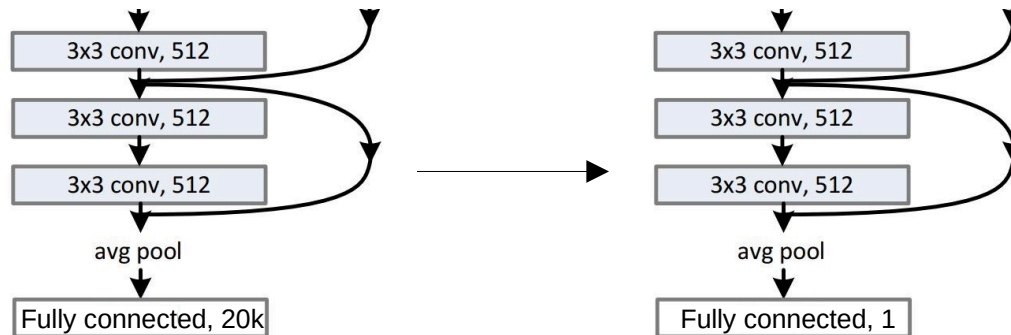
Idee

• Schritt 1

- Trainieren von tiefen CNNs mit großen Datenmengen
- Beispiel: ImageNet – 14 Millionen Bilder, 20000 Kategorien

• Schritt 2

- Ersetzen der Output Schicht
- Beispiel: Krebserkennung → logistische Regression (einzelnes Sigmoid Neuron)
- Training des letzten Layers (oder der letzten Layer) mit wenigen spezifischen Daten

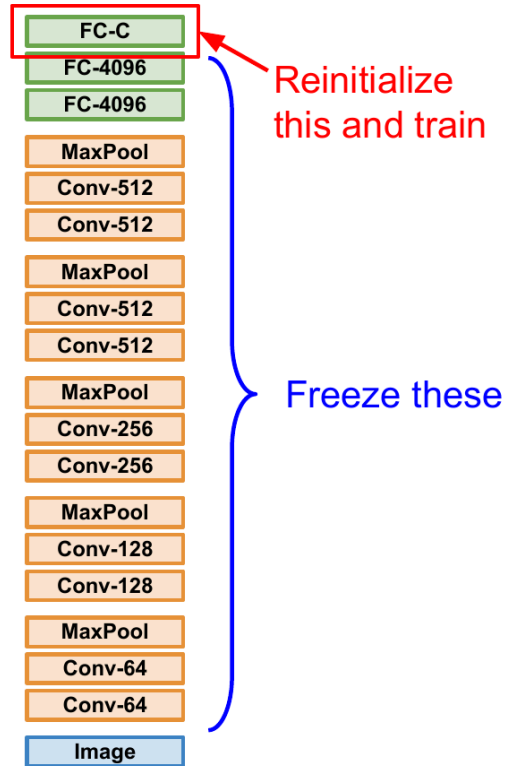


Transfer Learning

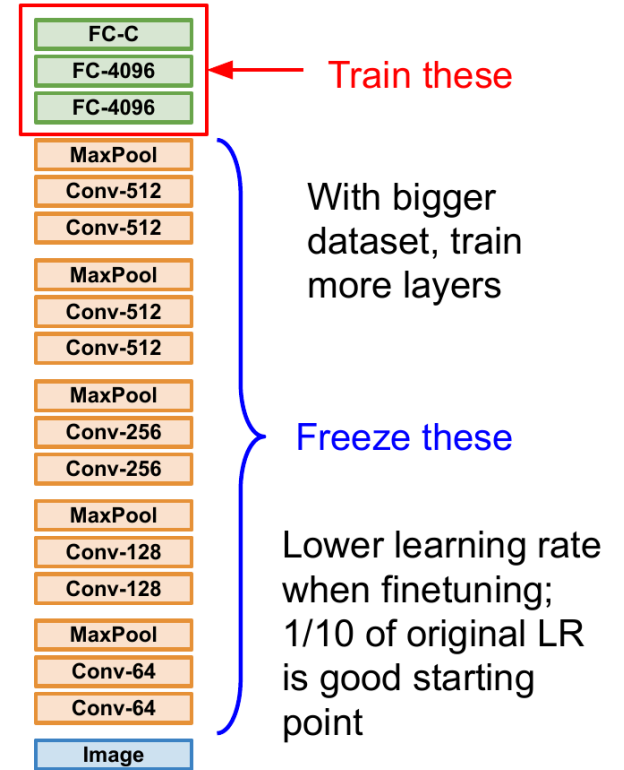
1. Train on Imagenet



2. Small Dataset (C classes)



3. Bigger dataset



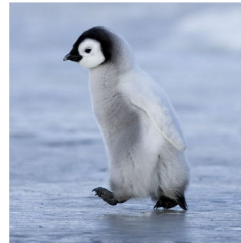
Datenerweiterung („Data augmentation“)

Ausnutzen von Label-Invarianzen

- CNNs sind nicht invariant gegenüber Rotationen
- CNNs können leicht invariant gegenüber Verschiebungen und Skalierungen sein (wegen Pooling)
- Viele (Klassifikations-)Labels sind invariant!

→ Data augmentation

- Verformung
- Skalierung
- Verschiebung
- Spiegelung



Original



Augmented

Teil 4

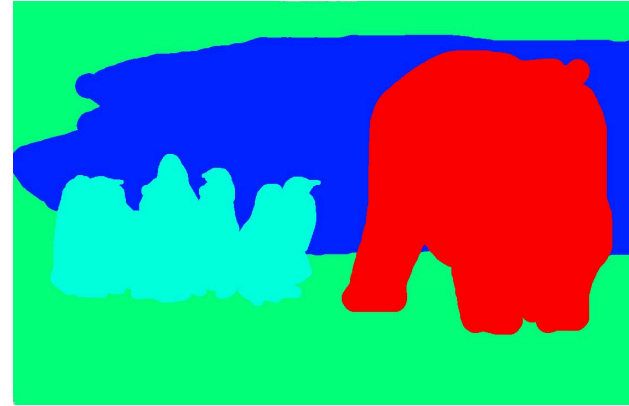
- 1) Bilddaten
- 2) Convolutional Neural Networks für Bildklassifizierung
- 3) Anwendungen, ResNet, Transfer Learning
- 4) Weitere Tasks: Objekterkennung, Segmentierung, ...**

Computer Vision Tasks

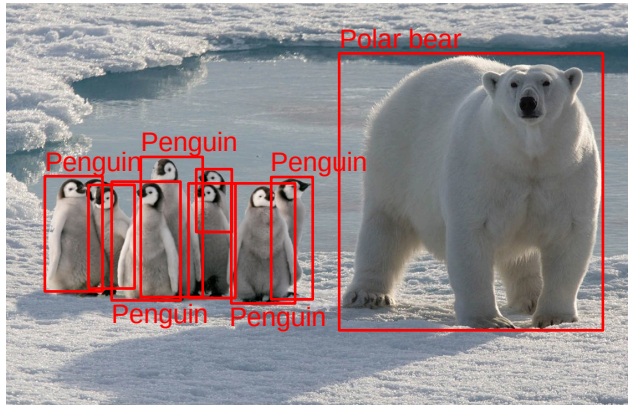
Klassifikation



Semantische Segmentierung



Objekterkennung



Instance segmentation



Semantische Segmentierung

Input: Bild, $n \times m$ Pixel

Output: Ein Label pro Pixel, $n \times m$ Klassifizierungen



Einzelner Pixel

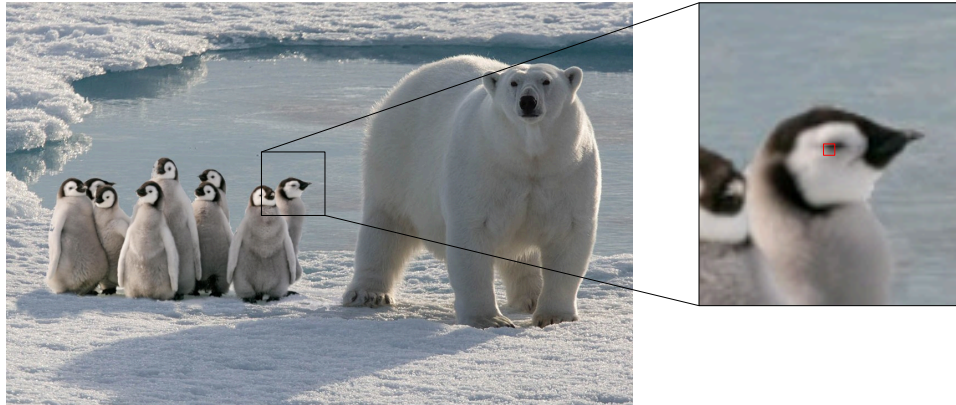
Pinguin/Bär/Wasser/Schnee?

→ **Unmöglich zu entscheiden**

Semantische Segmentierung

Input: Bild, $n \times m$ Pixel

Output: Ein Label pro Pixel, $n \times m$ Klassifizierungen



Pixel mit Umgebung (Kontext)

Pinguin/Bär/Wasser/Schnee?

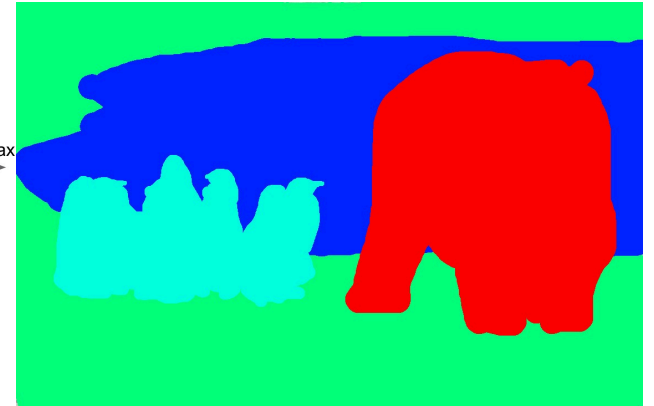
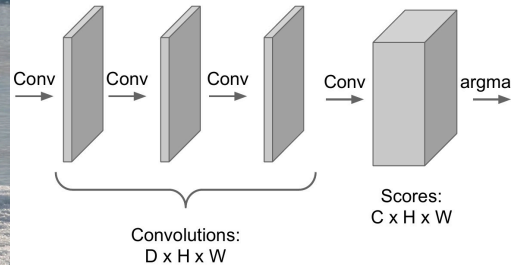
→ **Möglich, aber sehr ineffizient**

(muss für jeden Pixel einzeln gemacht werden)

Semantische Segmentierung

Input: Bild, $n \times m$ Pixel

Output: Ein Label pro Pixel, $n \times m$ Klassifizierungen



Idee

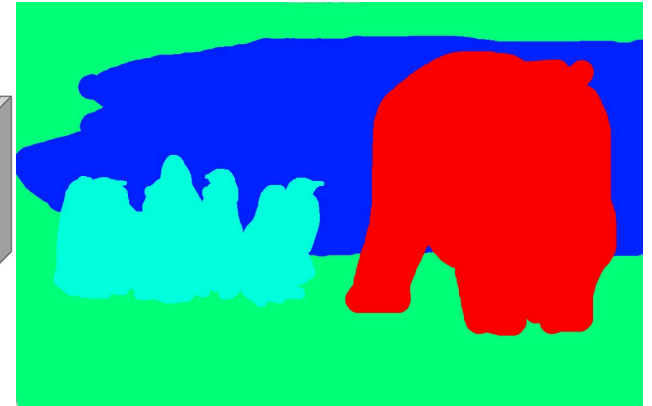
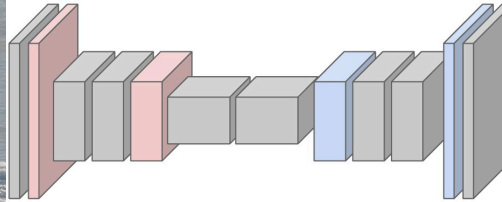
Ausschließlich Convolutional Layers

- Konstante Größe
- Eine Klassifikation pro Pixel am Ende
- Problem: Sehr teuer!

Semantische Segmentierung

Input: Bild, $n \times m$ Pixel

Output: Ein Label pro Pixel, $n \times m$ Klassifizierungen



Idee

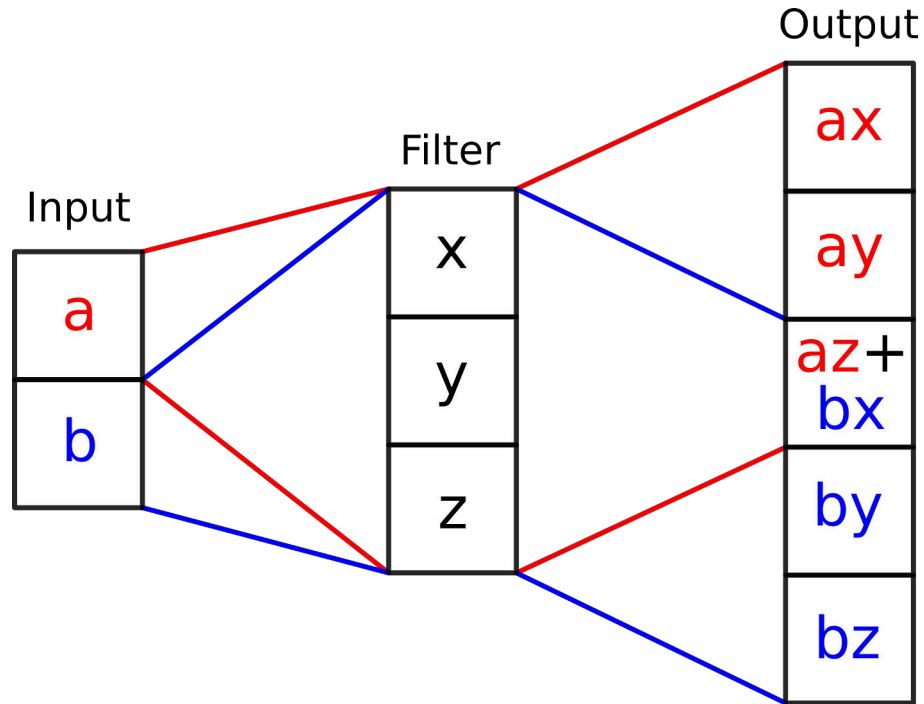
Ausschließlich Convolutional Layers

- Zuerst „downsampling“, dann „upsampling“
- Eine Klassifikation pro Pixel am Ende
- Wie funktioniert „Upsampling“?

„Upsampling“ → „Inverse Convolution“

Layer zum **Erhöhen** der Auflösung

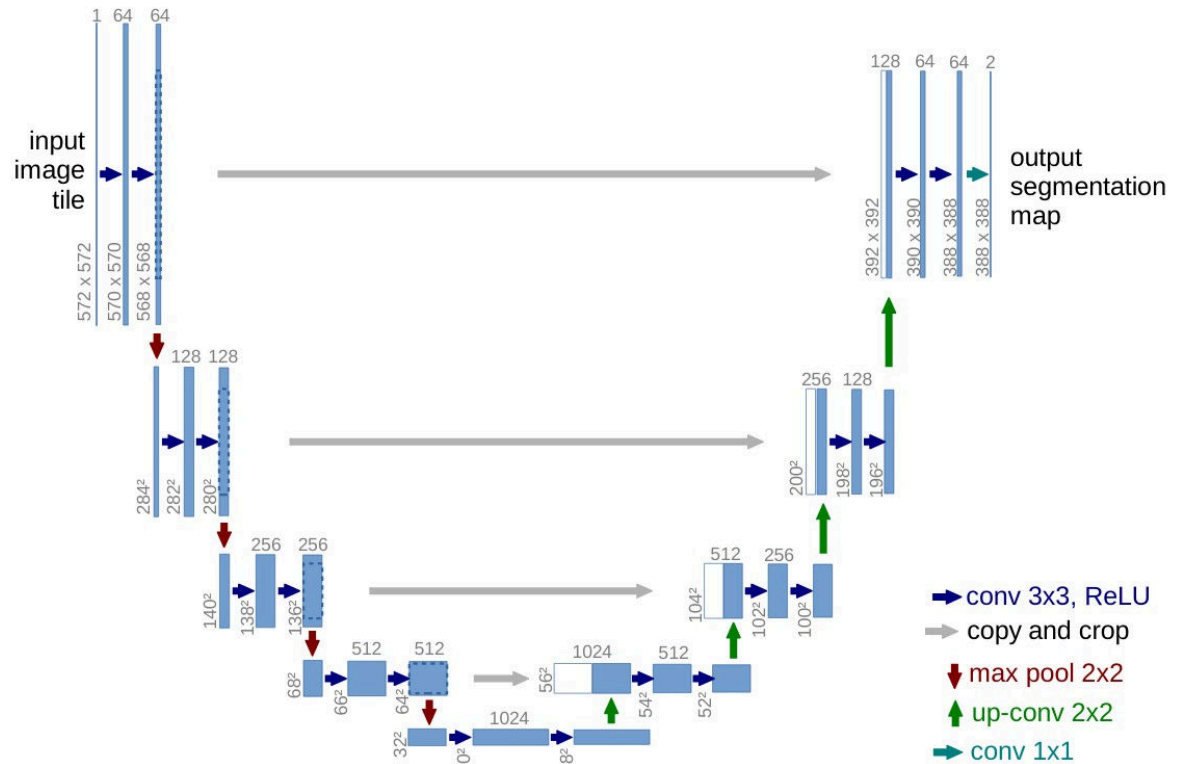
- Beispiel: 1D, Filtergröße 3, Stride 2



U-Net Modell

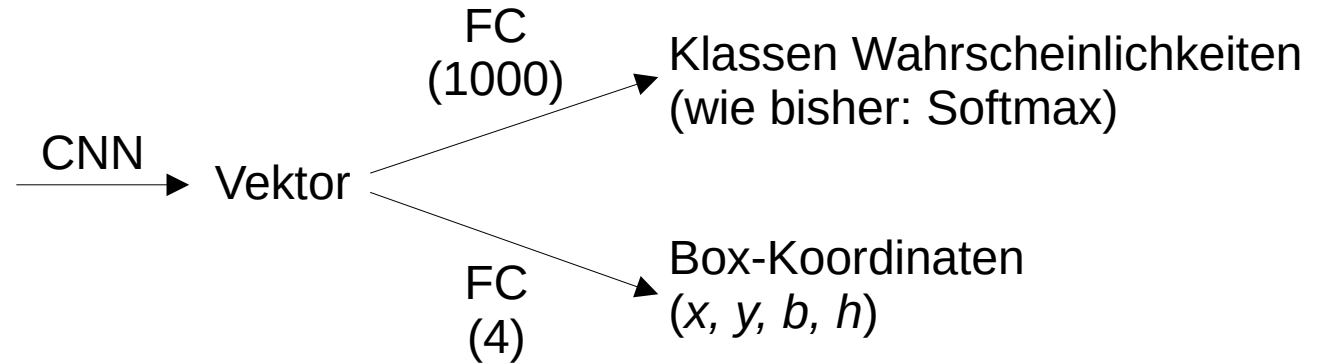
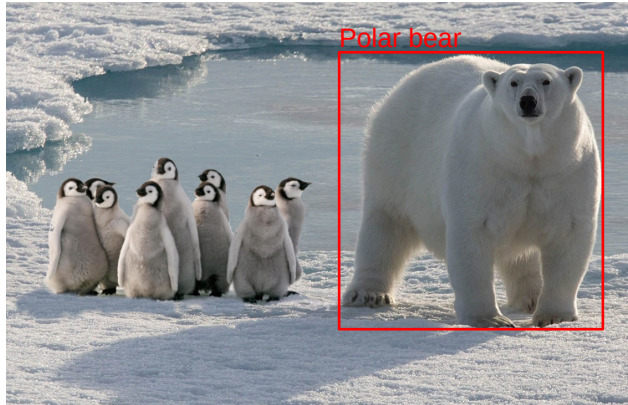
Architektur

- Bild → Bild (Grid → Grid)
- Schlüsselbestandteile
 - Convolutions
 - Pooling
 - Upconvolutions
 - Querverbindungen
- Benutzt für semantische Segmentierung



arXiv:1505.04597

Objekterkennung: Motivation



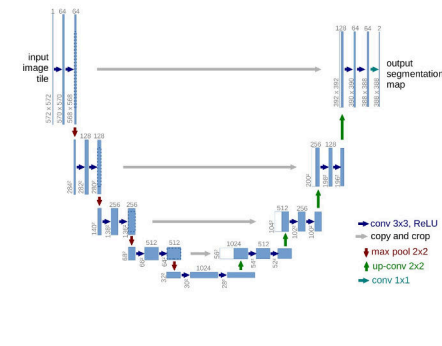
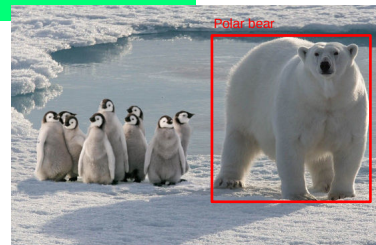
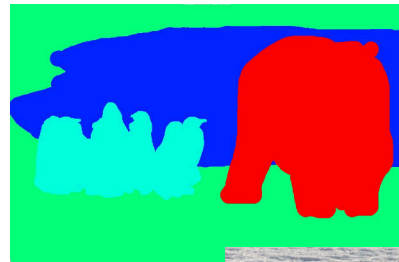
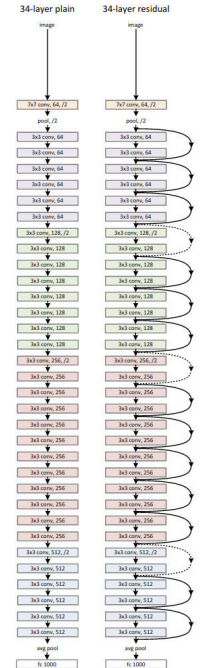
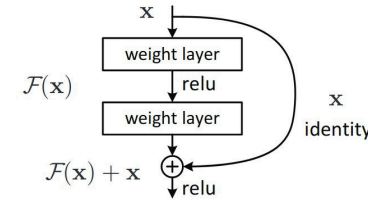
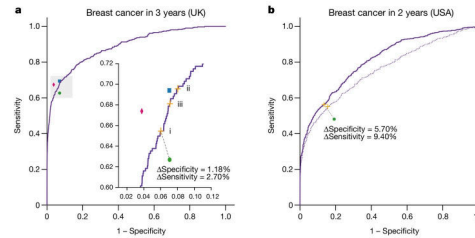
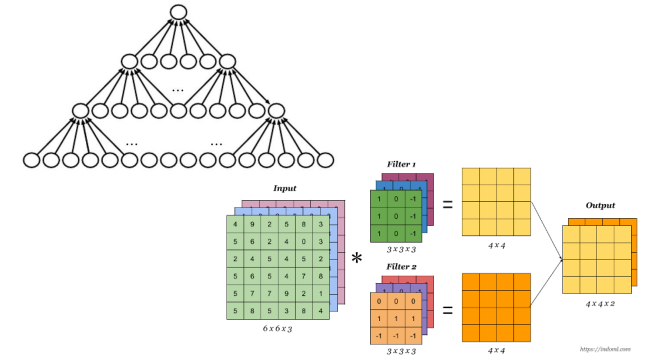
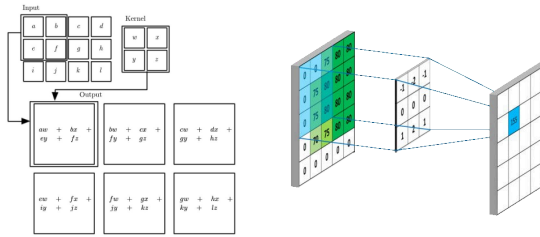
Zusammenfassung

- **Convolutional neural networks**

- Motivation
- Aufbau und Funktion
- Weight sharing and sparse connectivity
- Anzahl der Parameter in CNNs
- Pooling, padding, stride

- **Anwendungen, ResNet, Transfer Learning**

- Medizinische Bildanalyse
- Residual connections, ResNet
- Pretraining und Finetuning
- **Beyond Bildklassifizierung**
- Semantische Segmetierung
- U-Net
- Objekterkennung



Zusammenfassung

- Was ist ein CNN? Was sind die Komponenten und wie ist es aufgebaut?
- Wie viele Parameter hat ein conv layer mit gegebenen Spezifikationen?
- Warum funktionieren fully connected layers für Bild-Tasks nicht?
- Welche Hyperparameter hat ein CNN?
- Was sind residual connections und wofür werden sie verwendet?
- Was ist Transfer Learning?
- Was ist data augmentation? Was sind Invarianzen in diesem Kontext?
- Was sind mögliche Computer Vision Tasks?
- Was ist semantische Segmentierung?
- Was ist das U-Net und wofür wird es verwendet?
- Wie funktioniert Objekterkennung?