

Bildererkennung

Künstliche Intelligenz in der Bildverarbeitung:
Deep Learning & CNN
zur Erkennung von Bildinhalten

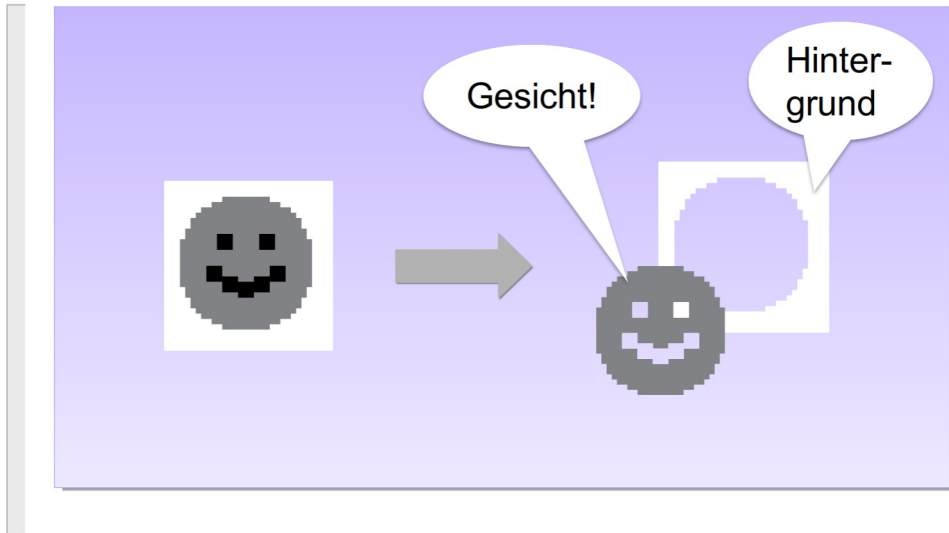
Wolfgang Heiden © 2021-22

Wolfgang Heiden © 2021-22

wolfgang.heiden@h-brs.de

Fachbereich Informatik (Dpt. Computer Science)
Hochschule Bonn-Rhein-Sieg – University of Applied Sciences,
53754 Sankt Augustin
Germany

Aufgabe: Erkennung von Bildinhalten



Stadien der automatisierten digitalen Bildverarbeitung

- **Bildbearbeitung**

- Punktoperationen, z.B. Kontrastanpassung
- Tiefpass- & Hochpassfilter
- Morphologische Operatoren

- **Bildaufbereitung**

- Clustering
- Segmentierung
- Skelettierung

- **Bildanalyse**

- Bildobjekte vs. Hintergrund
- Bilderkennung → KI
- Schrifterkennung (OCR), Gesichtserkennung, etc.

OCR = Optical Character Recognition

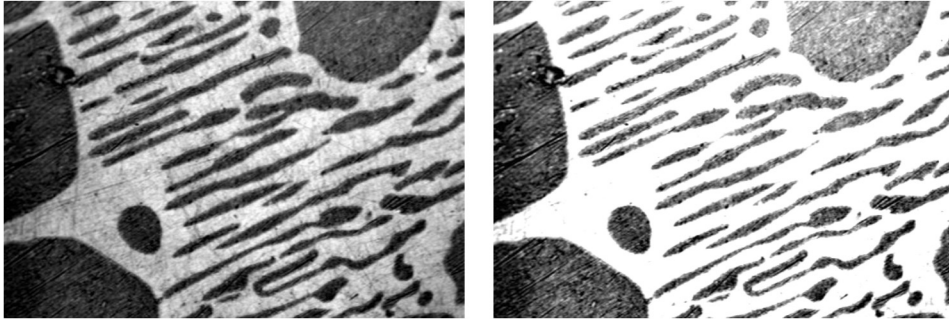
- **Vordergrund/Hintergrund**
- **Kanten um Bildobjekte**
- **Template Matching (Filter, MO)**
- **Segmentierung → Bildbereiche → Bildobjekte**
- **Skelettierung → Formen; z.B. OCR**
- **Mustererkennung**
- **Hit&Miss, Morphol. Filter → Formvergleich**
- **(D)(R)(C)NN**

MO = Morphologische Operationen (z.B. Hit&Miss)

Deep Recurrent Convolutional Neural Networks

Kontrastanpassung durch Grauwertspreizung → Homogenisierung des Hintergrunds

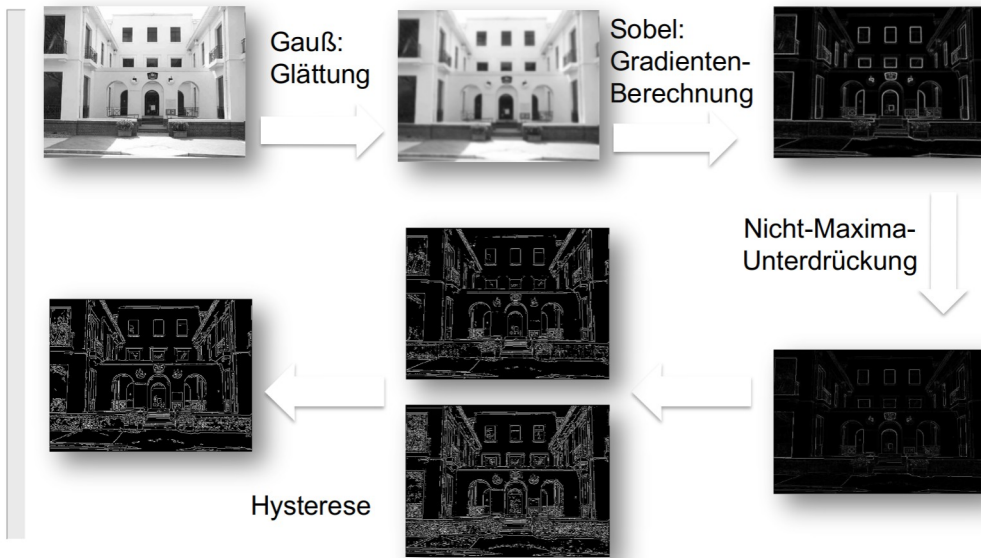
$$i_{new} = b_{low} + \frac{(b_{high} - b_{low})}{(a_{high} - a_{low})} \cdot (i - a_{low}); \quad a_{low} \leq i \leq a_{high}$$



$[a_{low}, a_{high}]$ = minimaler und maximaler Grauwert des ursprünglichen Bildes
 $[b_{low}, b_{high}]$ = minimaler und maximaler Grauwert des bearbeiteten Bildes

Vgl. Thema „Mathematische Bildanalyse“

Grenzen von Bildobjekten: Canny-Kantenfilter

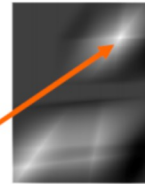


Template Matching mit Kreuzkorrelationsfiltern

● Template Matching



Umgebende Bounding Box ist
1 pixel größer als das Template



Maximum ist im
Mittelpunkt der
gesuchten Struktur.

Als Maßstab dafür, ob das Template zu einer Bildregion passt, kann der quadratische Abstand von Eingangsbild S_e und Template T verwendet werden:

$$q_s(x, y) = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} [S_e(x + p, y + q) - T(p, q)]^2$$

Vgl. Objektbasierte Segmentierung

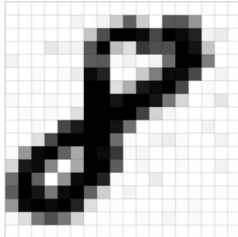
Aktuell: Objekterkennung mit Deep Learning

- **Classic AI vs. "Deep" AI**
- **Deep Convolutional Neural Networks**
 - CNN
- **Beispiel: Texterkennung**
 - OCR

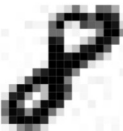
nach: Adam Geitgey (2016): Machine Learning is Fun!

<https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>

Zahl "8" (18x18px) als Datenstrom



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 12 0 11 39 137 37 0 152 147 84 0 0 0
0 0 1 0 0 0 41 160 250 255 235 162 255 238 206 11 13 0
0 0 0 16 9 9 158 251 45 21 184 159 154 255 233 40 0 0
10 0 0 0 0 0 145 146 3 10 0 11 124 253 255 187 0 0
0 0 3 0 4 15 236 216 0 0 38 189 247 240 169 0 11 0
1 0 2 0 0 0 233 253 23 62 124 241 255 164 0 5 0 0
6 0 0 4 0 3 252 250 228 255 255 234 112 28 0 2 17 0
0 2 1 4 0 12 225 253 253 255 172 31 0 0 1 0 0 0
0 0 4 0 163 225 251 255 229 120 0 0 0 0 0 11 0 0
0 0 21 162 255 255 254 255 126 6 0 10 14 6 0 0 9 0
3 79 242 255 141 66 255 245 189 7 8 0 0 5 0 0 0 0
26 221 237 98 0 87 251 255 144 0 0 0 0 7 0 0 11 0
133 235 141 0 87 184 255 100 3 0 0 13 0 1 0 1 0 0
145 248 228 116 235 255 141 34 0 11 0 1 0 0 0 1 3 0
85 237 253 246 255 210 21 1 0 1 0 0 6 2 4 0 0 0
6 23 112 157 114 32 0 0 0 0 2 0 0 0 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 11, 39, 137, 37, 0, 152, 147, 84, 0, 0, 0, 0,
0, 1, 0, 0, 0, 41, 160, 250, 255, 235, 162, 255, 238, 206, 11, 13, 0, 0, 0, 16, 9, 9, 150, 251, 45, 21, 184, 159, 154, 2
55, 233, 40, 0, 0, 10, 0, 0, 0, 0, 145, 146, 3, 10, 0, 11, 124, 253, 255, 187, 0, 0, 0, 3, 0, 4, 15, 236, 216, 0, 0,
38, 189, 247, 240, 169, 0, 11, 0, 1, 0, 2, 0, 0, 253, 253, 23, 62, 124, 241, 255, 164, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3, 252
, 250, 228, 255, 255, 234, 112, 28, 0, 2, 17, 0, 0, 2, 1, 4, 0, 21, 255, 253, 251, 255, 172, 31, 0, 0, 1, 0, 0, 0, 0, 4,
0, 163, 225, 251, 255, 229, 120, 0, 0, 0, 0, 11, 0, 0, 0, 21, 162, 255, 255, 254, 255, 126, 6, 0, 10, 14, 6, 0, 0, 9
, 0, 3, 79, 242, 255, 141, 66, 255, 245, 189, 7, 8, 0, 0, 5, 0, 0, 0, 26, 221, 237, 98, 0, 67, 251, 255, 144, 0, 8, 0, 0
, 7, 0, 0, 11, 0, 125, 255, 141, 0, 87, 244, 255, 208, 3, 0, 0, 13, 0, 1, 0, 1, 0, 0, 145, 248, 228, 116, 235, 255, 141, 34
, 0, 11, 0, 1, 0, 0, 0, 1, 3, 0, 85, 237, 253, 246, 255, 210, 21, 1, 0, 1, 0, 0, 6, 2, 4, 0, 0, 0, 6, 23, 112, 157, 114, 32
, 0, 0, 0, 0, 2, 0, 8, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Quelle: Geitgey: ML is Fun!, 2016

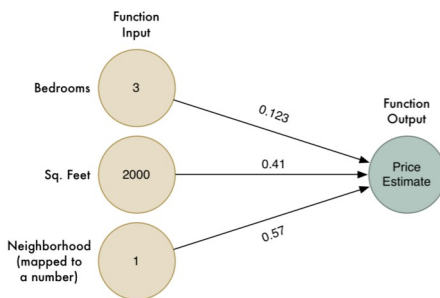
- Machine Learning
 - Supervised vs. Unsupervised
- Gewichtungsfaktoren
 - = Multiplikatoren für einzelne Parameter/Attribute
 - Kostenfunktion: Fehler im Ergebnis bei bisheriger Gewichtung

Based on: Adam Geitgey (2016): Machine Learning is Fun!

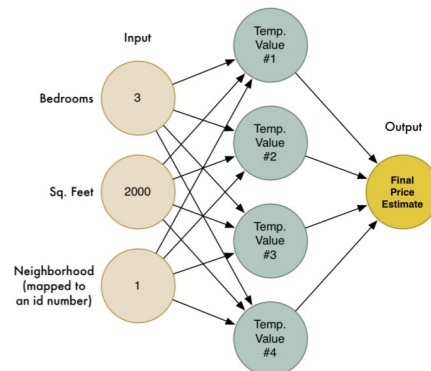
<https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>

Preisschätzung über ein NN

```
def estimate_house_sales_price(num_of_bedrooms, sqft, neighborhood):
    price = 0 # a little pinch of this
    price += num_of_bedrooms * 0.123 # and a big pinch of that
    price += sqft * 0.41 # maybe a handful of this
    price += neighborhood * 0.57
    return price
```



Quelle: Geitgey: ML is Fun!, 2016



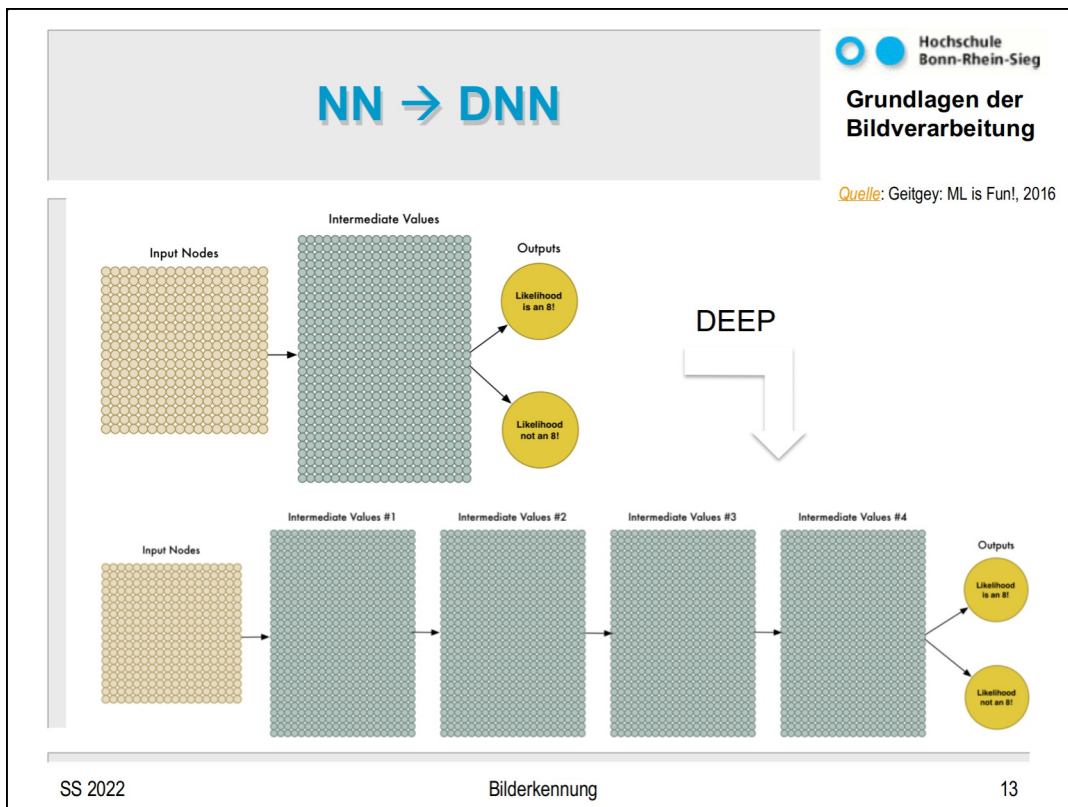
Als Beispiel für ein einfaches Neuronales Netz (NN) wird eine Preisabschätzung herangezogen.

- Risiko Overfitting
- Parametrisierung
 - Geeignete Daten finden (tatsächliche Beziehungen)
- Recurrent Neural Network (RNN) → sich beim Gebrauch kontinuierlich anpassendes/aktualisierendes NN
- Neurone
 - Input Layer + Hidden Layer + Output Layer
- Einfache Bilderkennung über NN
 - Azentrische Positionen (Lageinvarianz) → *Fehler*

Wird es einem NN erlaubt, sich bestmöglich an eine einzelne Datenquelle anzupassen, besteht ein Risiko für „Overfitting“. D.h. Das System hat sich auf genau die Trainingsdaten hin optimiert und kann bei leicht abweichenden Daten aus anderer Quelle versagen.

Das kann insbesondere dann passieren, wenn zufällige Übereinstimmungen zwischen Datensätzen derselben Herkunft als Kategorisierungskriterien identifiziert werden. Daher ist es wichtig, bei der Parametrisierung eines NN tatsächliche Abhängigkeiten bzw. ursächliche Zusammenhänge zugrunde zu legen.

Einfache Bilderkennungsverfahren können bei der Erkennung von Ähnlichkeiten bereits versagen, wenn das zu identifizierende Muster durch Translation und/oder Rotation gegenüber dem gelernten Vorbild verändert wurde.



Aus einem einfachen (herkömmlichen) NN wird durch Hinzufügen zahlreicher Zwischenschichten ein „tiefes“ DNN.

NN = Neural Network

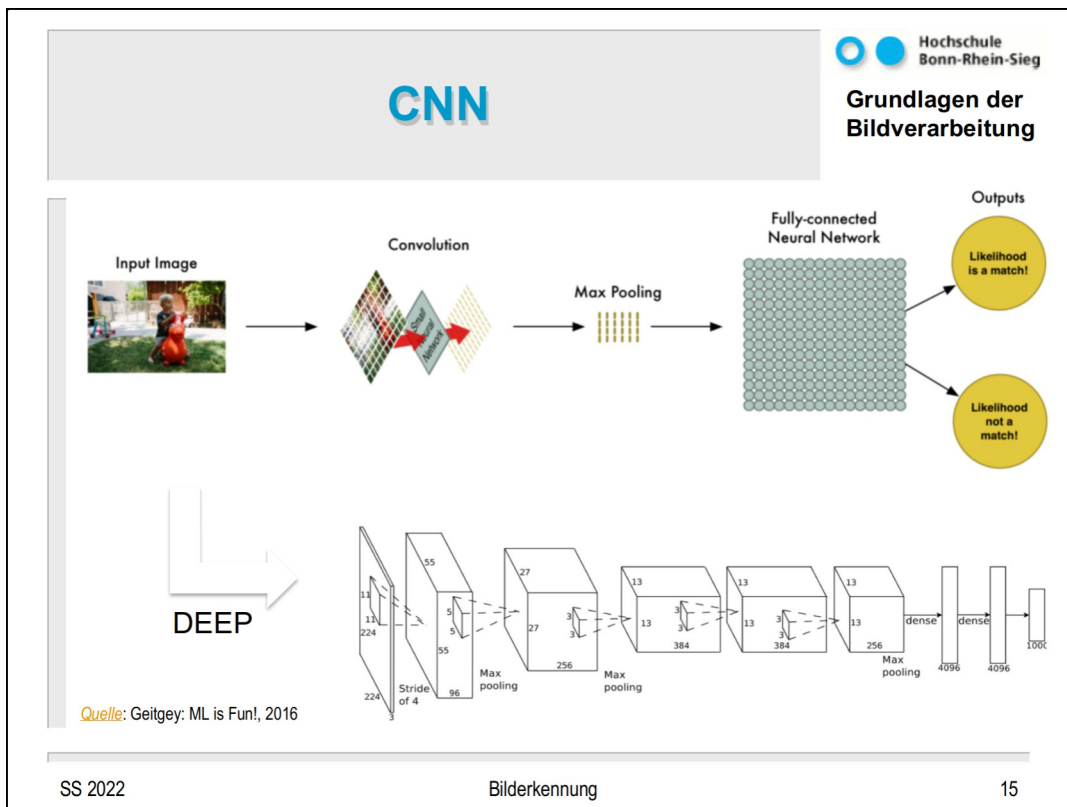
DNN = Deep Neural Network

- Convolution (Wicklung, ~Faltung)
 - Transformationsinvarianz gleicher Strukturen
 - Sliding window → overlapping picture tiles
 - Identical weights for all tiles
 - Tile grid → output grid
 - Downsampling: Verkleinerung des Output grids
 - 1 Max-pooling
 - Fully connected NN (layers)
- Use 3D graphics HW for DML calculations

NN = Neural Network

HW = Hardware

DML = Deep Machine Learning



Ein "Convolutional Neural Network" (CNN) kommt dadurch zustande, dass in einigen Zwischenschichten durch Mittelung (od. andere Auswahl von repräsentativen Werten für benachbarte Datengruppen) das Datenaufkommen verkleinert wird.

- Adam Geitgey (2016): **Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks**
 - <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>

Insgesamt: <https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>

● Fallstricke

- 95% accuracy → irreführende statistische Begriffe
 - 1 TP,TN,FP,FN (vgl. MedBV) → Precision vs. Recall (vgl. Canny-Hysteresis)
- Oversampling
- 1pix-Errors (vgl. Akhtar & Mian)

● Face recognition

- Facebook 98% accuracy
- Unintentional racist algorithms
- Face detection (HOG) – posing&projecting (warping face landmarks) – DCNN training on identical & different persons – embeddings – relate data to names (SVM)

Wird der Begriff „Accuracy“ nicht näher definiert, dann kann er leicht missverstanden werden.

Hier gilt es im Einzelfall zwischen Genauigkeit (Precision, d.h. kaum FP) und Vollständigkeit (Recall, d.h. kaum FN) zu unterscheiden. Dabei spielt das Verhältnis richtig oder falsch zugeordneter Datensätze eine wesentliche Rolle:

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

ML-Systeme zur Gesichtserkennung, die nur oder hauptsächlich mit Bildern aus spezifischen ethnischen Gruppen trainiert wurden, neigen bei Gesichtern aus anderen ethnischen Gruppen oft zu krassen Fehleinschätzungen.

Akhtar und Mian konnten nachweisen, dass in ML-Systemen nach Oversampling bereits Änderungen eines einzelnen Pixels die richtige Erkennung gelernter Bildobjekte verhindern können.

HOG = Histogram of Oriented Gradients (Lage- und Orientierungsbeziehungen zwischen charakteristischen Gesichtsregionen, z.B. Augen, Nase und Mund)

DCNN = Deep Convolutional Neural Network

SVM = Support Vector Machine

Schwachstellen der KI-Bilderkennung mit DNN

 Hochschule
Bonn-Rhein-Sieg

**Grundlagen der
Bildverarbeitung**

- Advantages of organic intelligence
 - Flexibility
 - Versatility
- Risk of **Overfitting** in Deep Learning



Akhtar & Mian: Adversarial Attacks on Deep Learning, J LaTeX Class Files, 2017

SS 2022

Bilderkennung
Hypermedia in Academic Education (W. Heiden, 2018)

18

Image recognition: paper MAS graduate N. Akhtar:

Naveed Akhtar & Ajmal Mian: Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. J. LaTeX Class Files, Aug. 2017

Example for corruption of AI

Kompetenzcheck

- **Vorverarbeitung mit verschiedenen Bildverarbeitungskonzepten**
 - Punktoperationen, Kreuzkorrelationsfilter
- **Einfache Ansätze zur Bilderkennung**
 - Template Matching, Morphologische Operatoren
- **Maschinelle Lernverfahren**
 - Neuronale Netze
 - CNN
 - Deep Machine Learning (DML)
 - Grenzen der Bilderkennung mit DML

