

Bin Packing Problem: A general purpose Hill Climbing procedure

Lukas Schmauch, Sebastian Wolf

Seminar Modern Heuristics
Dr. Rico Walter

Februar 2021

Übersicht

Was ist das Bin Packing Problem?

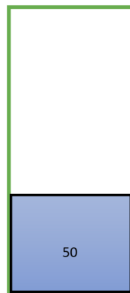
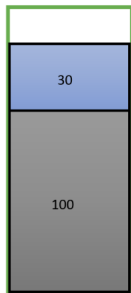
Hill Climbing Ansatz

Computational Studies

Zusammenfassung

Bin Packing Problem - Kurzübersicht

Grundidee Hill Climbing Ansatz



Ablauf Hill Climbing Verfahren

1. Eröffnungsverfahren (First Fit Descending)
2. Zufällige Auswahl von Bins
3. Ausführung des Verbesserungsverfahrens
4. Füge Gruppen wieder zusammen
5. Shuffle der Gruppen
6. Greedy Algorithmus
7. Wiederhole Schritt 2-6 bis Abbruchkriterium erreicht

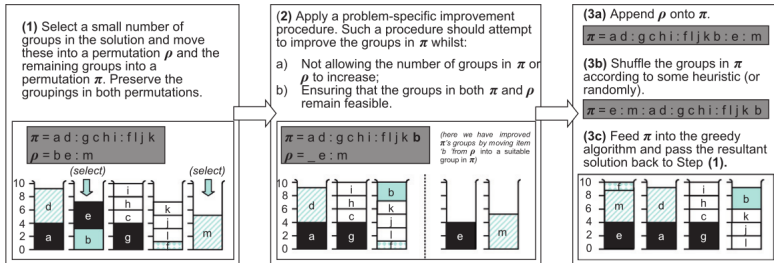


Figure: Greedy Algorithmus

1. Eröffnungsverfahren - First Fit Descending

► Sortierung nach absteigender Itemkapazität

```
GREEDY( $\pi, G$ )
(1)   for ( $i \leftarrow 1$  to  $|\pi|$ )
(2)      $j \leftarrow 1$ 
(3)     found  $\leftarrow$  false
(4)     while (not found and  $j \leq G$ )
(5)       if (Group  $j$  is feasible for Item  $\pi[i]$ )
(6)         Insert Item  $\pi[i]$  into Group  $j$ 
(7)         found  $\leftarrow$  true
(8)       else
(9)          $j \leftarrow j + 1$ 
(10)    if (not found)
(11)       $G \leftarrow G + 1$  /*A new group is opened*/
(12)      Insert  $\pi[i]$  into group  $G$ 
```

Figure: Greedy Algorithmus

► Bild mit resultierender Lösung

2. Zufällige Auswahl von Bins

3. Ausführung des Verbesserungsverfahrens

4. Zusammenfügen der Gruppen

5. Shuffle der Gruppen

6. Greedy Algorithmus

Ablauf Hill Climbing Verfahren

1. Eröffnungsverfahren (First Fit Descending)
2. Zufällige Auswahl von Bins
3. Ausführung des Verbesserungsverfahrens
4. Füge Gruppen wieder zusammen
5. Shuffle der Gruppen
6. Greedy Algorithmus
7. **Wiederhole Schritt 2-6 bis Abbruchkriterium erreicht**

3. Verbesserungsverfahren im Detail

BPP-IMPROVEMENT-PROCEDURE(π, ρ, C)

```
(1)  for ( $g \leftarrow 1$  to  $G(\pi)$ )
(2)    foreach (pair of items  $i, j$  in group  $g$  in  $\pi$ )
(3)      for ( $h \leftarrow 1$  to  $G(\rho)$ )
(4)        foreach (pair of items  $k, l$  in group  $h$  in  $\rho$ )
(5)           $\delta = s[k] + s[l] - s[i] + s[j]$ 
(6)          if ( $\delta > 0$  and  $F(g) + \delta \leq C$ )
(7)            Move items  $i$  and  $j$  into group  $h$  in  $\rho$  and move items  $k$  and  $l$  into group  $g$  in
               $\pi$ 
(8)    foreach (pair of items  $i, j$  in group  $g$  in  $\pi$ )
(9)      for ( $h \leftarrow 1$  to  $G(\rho)$ )
(10)        foreach (item  $k$  in group  $h$  in  $\rho$ )
(11)           $\delta = s[k] - s[i] + s[j]$ 
(12)          if ( $\delta > 0$  and  $F(g) + \delta \leq C$ )
(13)            Move items  $i$  and  $j$  into group  $h$  in  $\rho$  and move item  $k$  into group  $g$  in  $\pi$ 
(14)    foreach (item  $i$  in group  $g$  in  $\pi$ )
(15)      for ( $h \leftarrow 1$  to  $G(\rho)$ )
(16)        foreach (item  $k$  in group  $h$  in  $\rho$ )
(17)           $\delta = s[k] - s[i]$ 
(18)          if ( $\delta > 0$  and  $F(g) + \delta \leq C$ )
(19)            Move items  $i$  into group  $h$  in  $\rho$  and move item  $k$  into group  $g$  in  $\pi$ 
```

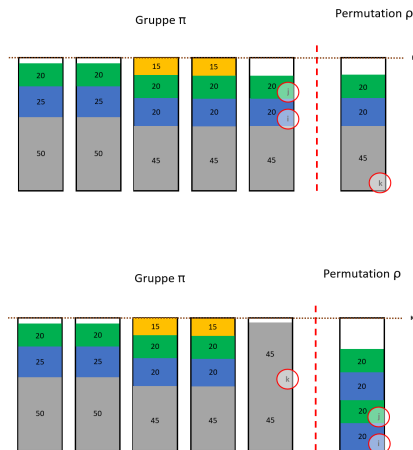
Figure: Verbesserungsverfahren Pseudocode

2:2 Move

- ▶ $\delta = s[k] + s[l] - s[i] - s[j]$
- ▶ $\delta = 4 + 3 - 2 - 1 = 4$
- ▶ *if* ($\delta > 0$ and $F(g) + \delta \leq C$)
- ▶ *if* ($4 > 0$ and $10 + 4 \leq 20$)

2:1 Move

- ▶ $\delta = s[k] - s[i] - s[j]$
- ▶ $\delta = 45 - 20 - 20 = 5$
- ▶ *if* ($\delta > 0$ and $F(g) + \delta \leq C$)
- ▶ *if* ($5 > 0$ and $85 + 5 \leq 91$)



1:1 Move

- ▶ $\delta = s[k] - s[i]$
- ▶ $\delta = 4 - 2 = 2$
- ▶ *if* ($\delta > 0$ and $F(g) + \delta \leq C$)
- ▶ *if* ($2 > 0$ and $10 + 2 \leq 20$)

Computational Studies - Genutzte Instanzen

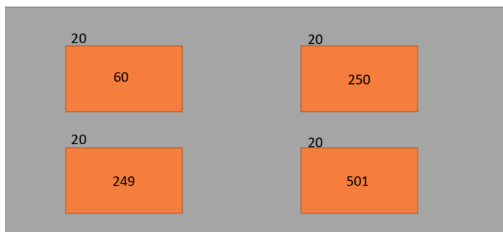


Figure: Instanzen Triplet

Übersicht der Ergebnisse

- ▶ $LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil$
- ▶ **Uniform:** $LB = z^*$ bei **79 von 80 Instanzen** (eine Instanz mit $z^* = LB + 1$)
- ▶ **Hard:** $LB = z^*$ bei **3 von 10 Instanzen** (Bei 7 von 10 $z^* = LB + 1$)
- ▶ **Triplet:** $LB = z^*$ für **alle Instanzen**

Typ	Inst.	#Items	Mittlere LB	FFD	HC	HC*	Mittlere Zeit
Unif	20	120	49.1	0.7	0.05	0.1	6.24
Unif	20	250	101.6	1.5	0.25	0.25	27.19
Unif	20	500	201.2	2.7	0.15	0.15	25.73
Unif	20	1000	400.6	4.85	0.25	0.2	42.91
Hard	10	200	55.5/56.2	4.1/3.4	0.8	0	81.04
Trip	20	60	20	3.2	1	0.85	100
Trip	20	120	40	5.8	1	1	100
Trip	20	249	83	12.1	1	1	100
Trip	20	510	167	23.05	1.1	1	100

Optimalitätsanalyse Instanzgruppe Uniform & Hard

$$LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil \quad (1)$$

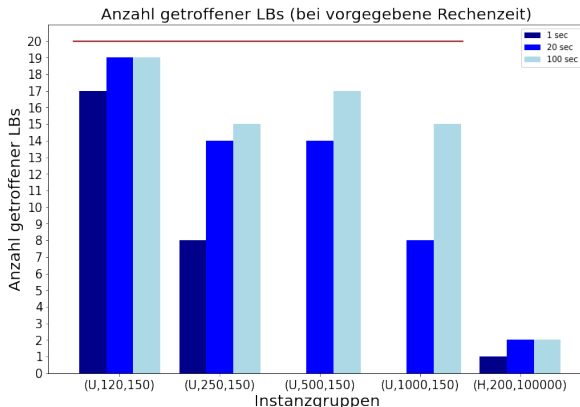


Figure: Anzahl getroffener LBs

Worst Case Analyse Instanzgruppe Uniform & Hard

$$LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil \quad (2)$$

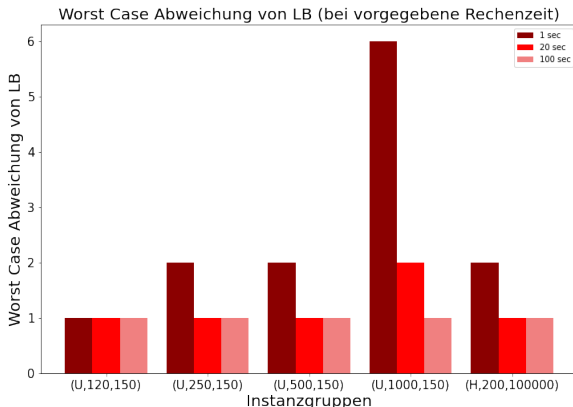


Figure: Worst Case Abweichung von LB

Lösungsgüte im Zeitverlauf Instanzgruppe Uniform & Hard

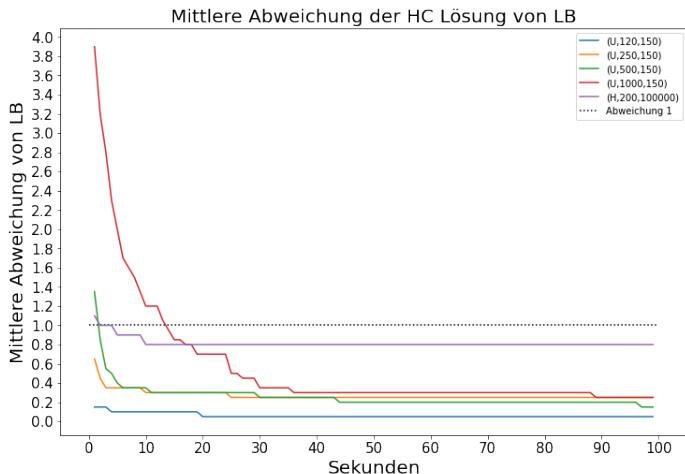


Figure: Mittlere Abweichung von LB pro Zeiteinheit

Optimalitätsanalyse Instanzgruppe Triplet

- ▶ $LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil = \frac{\#Items}{3}$
- ▶ LB wird nie getroffen

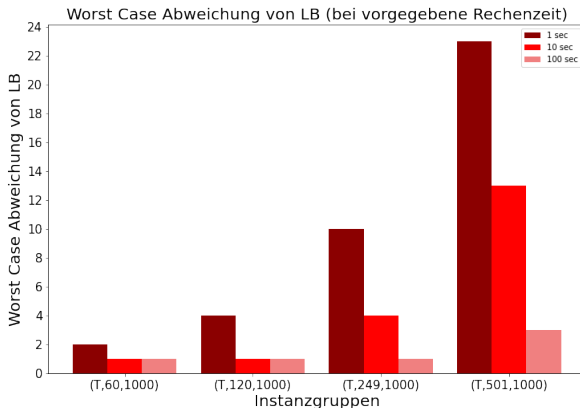


Figure: Worst Case Abweichung von LB

Lösungsgüte im Zeitverlauf Instanzgruppe Triplet

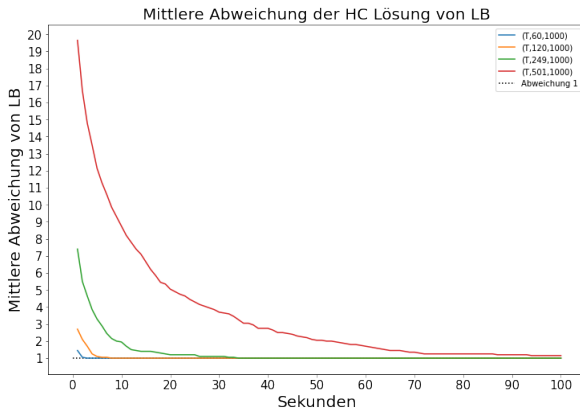


Figure: Mittlere Abweichung von LB pro Zeiteinheit

Vergleich Uniform, Hard und Triplet

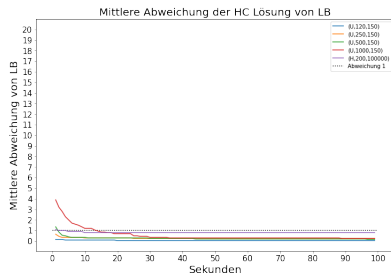


Figure: Uniform und Hard

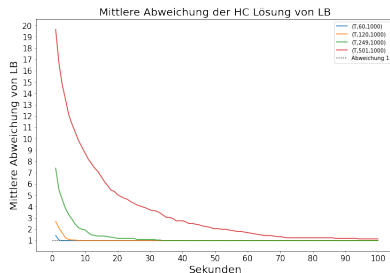


Figure: Triplet

Verteilungsfunktion

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (3)$$

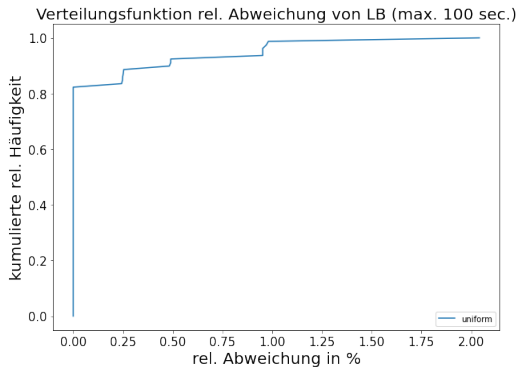


Figure: rel. Abweichung von LB

Verteilungsfunktion

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (4)$$

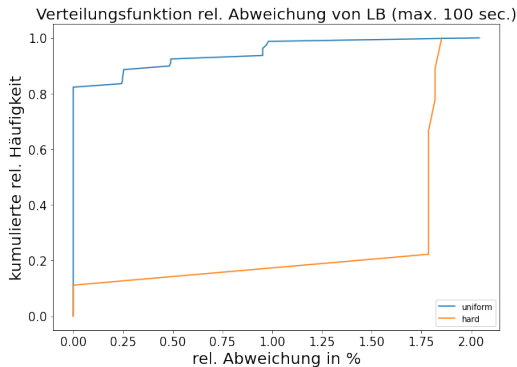


Figure: rel. Abweichung von LB

Verteilungsfunktion

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (5)$$

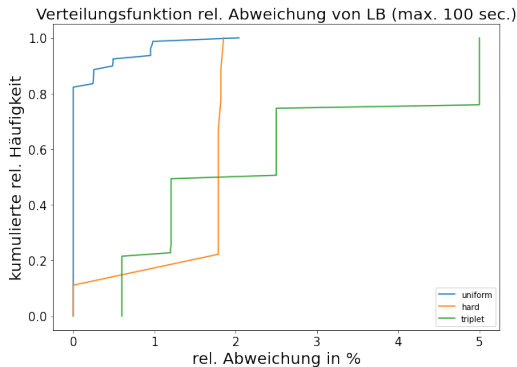


Figure: rel. Abweichung von LB

Verteilungsfunktion - Verfahrensänderung (Uniform)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (6)$$

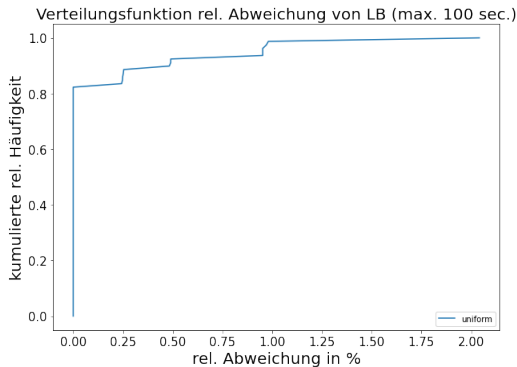


Figure: rel. Abweichung von LB

Verteilungsfunktion - andere Permutationswahl

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (7)$$

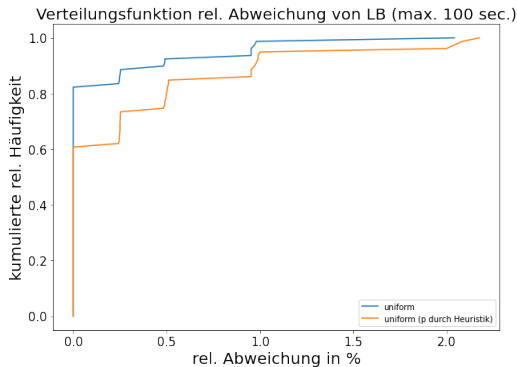


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (8)$$

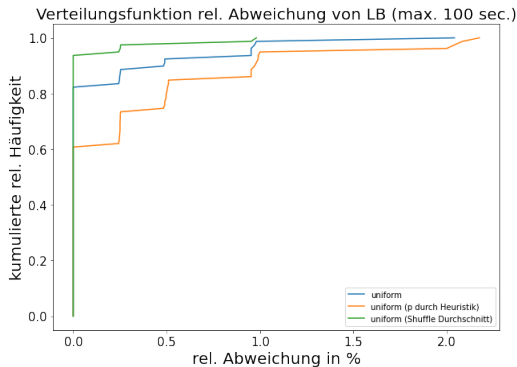


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (inkl. mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (9)$$

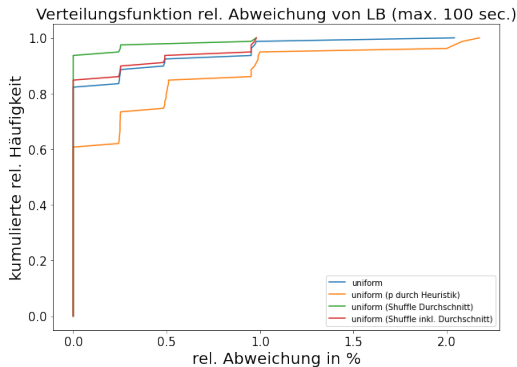


Figure: rel. Abweichung von LB

Verteilungsfunktion - Verfahrensänderung (Triplet)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (10)$$

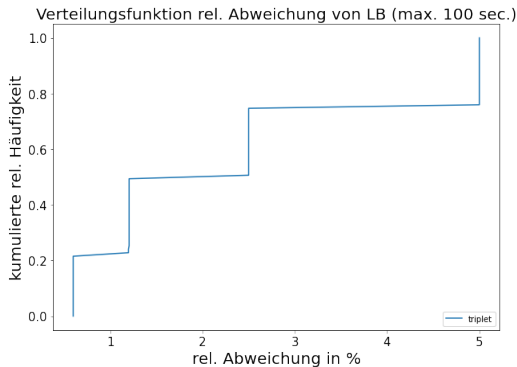


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (11)$$

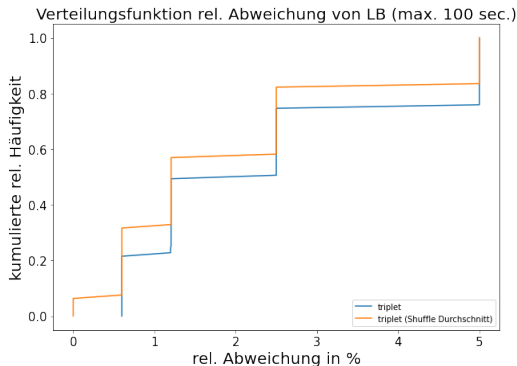


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (inkl. mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (12)$$

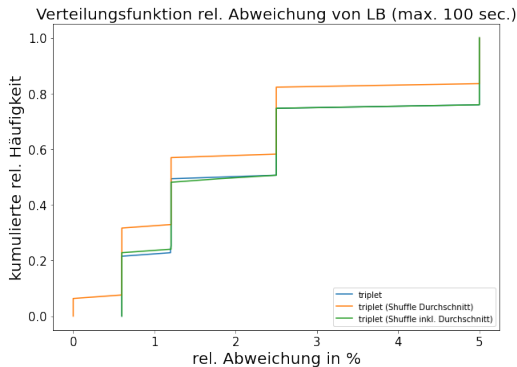


Figure: rel. Abweichung von LB

Zusammenfassung und Ausblick

- ▶ Weitere Untersuchungen des Shuffle-Operators
- ▶ Weglassen von Moves

Triplet: Vergleich mit Shuffle (mittlere Itemkapazität)

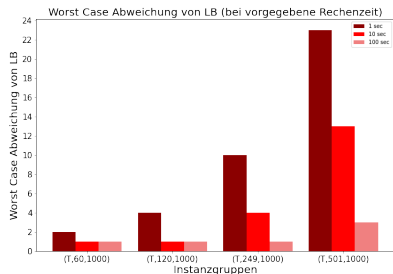


Figure: Standardverfahren

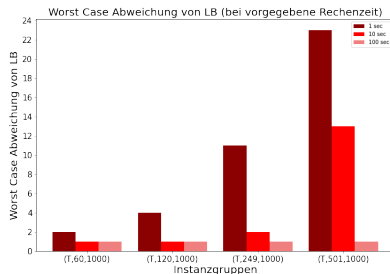


Figure: mittlere Itemkapazität

Uniform: Vergleich mit anderer Permutationswahl

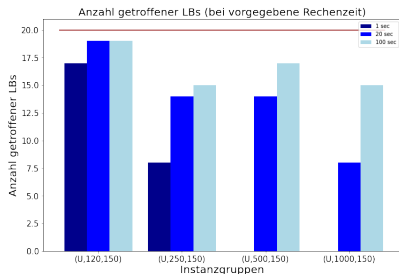


Figure: Random Permutation

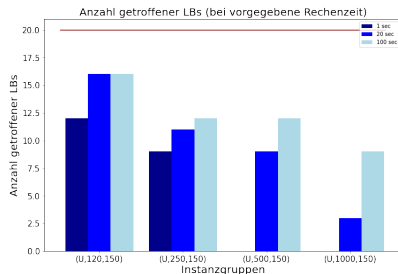


Figure: Minimale Itemzahl

Literaturverzeichnis



Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, 2016



Convolutional neural networks: an overview and application in radiology, Yamashita, 2018



Script: Einführung in tiefe Lernverfahren - Faltungsnetzwerke,
Prof. Joachim Denzler



[https://towardsdatascience.com/
a-comprehensive-guide-to-convolutional-neural-networks\
-the-eli5-way-3bd2b1164a53](https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)



<https://aishack.in/tutorials/image-convolution-examples/>