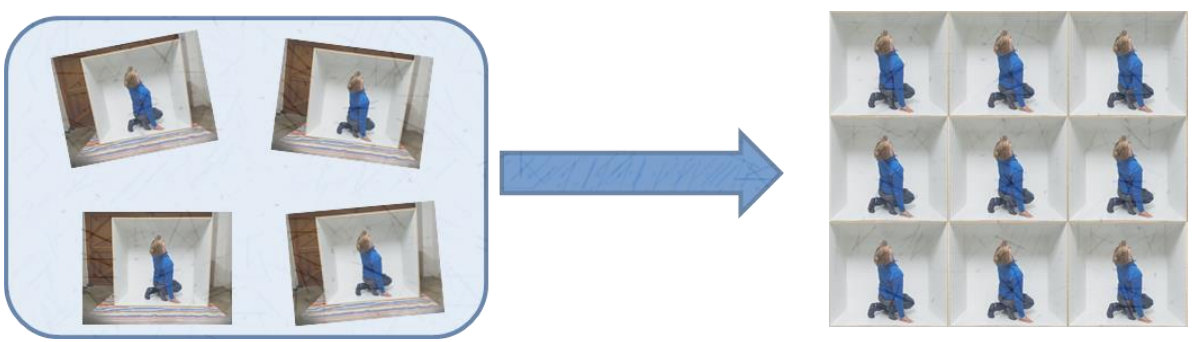
Digitale Bildverarbeitung 1

Einführung in die digitale Bilderverarbeitung für Informatikstudierende im Bachelor

Vorlesung: Michael Möller – michael.moeller@uni-siegen.de
Übungen: Hannah Dröge – hannah.droege@uni-siegen.de



Filter - Punktweise

Wie können wir die Qualität von Bildern verbessern?

Das hängt von der Art der Bilddefekte ab!

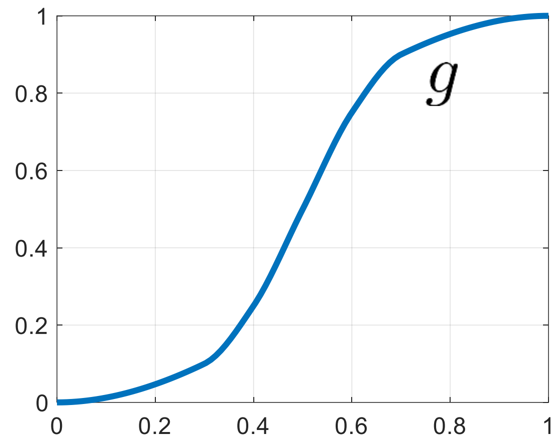
Beginnen wir mit niedrigem Kontrast

Erinnerung
- Schwarz: Pixelwert 0
- Weiß: Pixelwert 1

Problem: Die meisten Werte sind in der Nähe von 0.5. Es gibt wenig wirklich große oder kleine Werte.

Filter - Punktweise

Möglichkeit: „Ziehe die Werte des Bildes auseinander“



$$f_{ij}^{new} = g(f_{ij})$$

An jedem Pixel,

ersetze den aktuellen

Wert f_{ij} durch einen

Wert $g(f_{ij})$, wobei

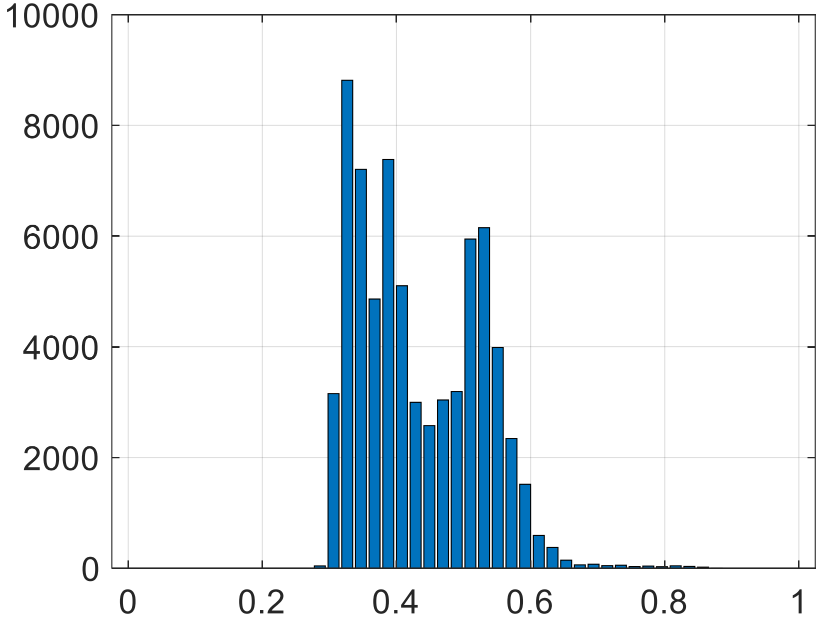
S-förmig ist.

Filter - Punktweise

Kann man automatisch erkennen, ob Pixelwerte auseinandergezogen werden sollten?

Wertvolle Information: ***Histogramm***

Anzahl der Pixel ...

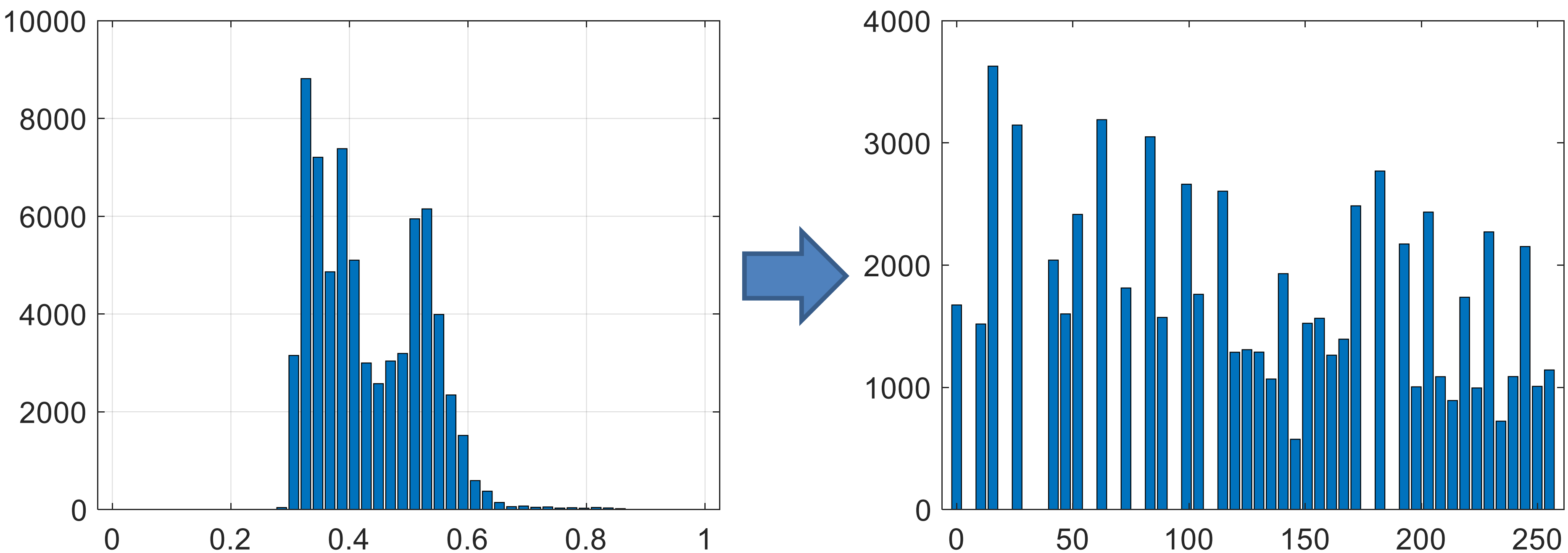


... die einen Wert in einem bestimmten Intervall haben. Dies sind die Mittelpunkte der Intervalle

Filter - Punktweise

Mögliche Verwendung des Histogramms: Bestimme die Transformation g so, dass das Histogramm des Bildes nach Anwenden von g möglichst einem gewünschten Histogramm entspricht. (Oft: Optimierungsproblem)

$$\min_g \|\text{hist}(g(f_{ij})) - \text{hist}_{goal}\|$$



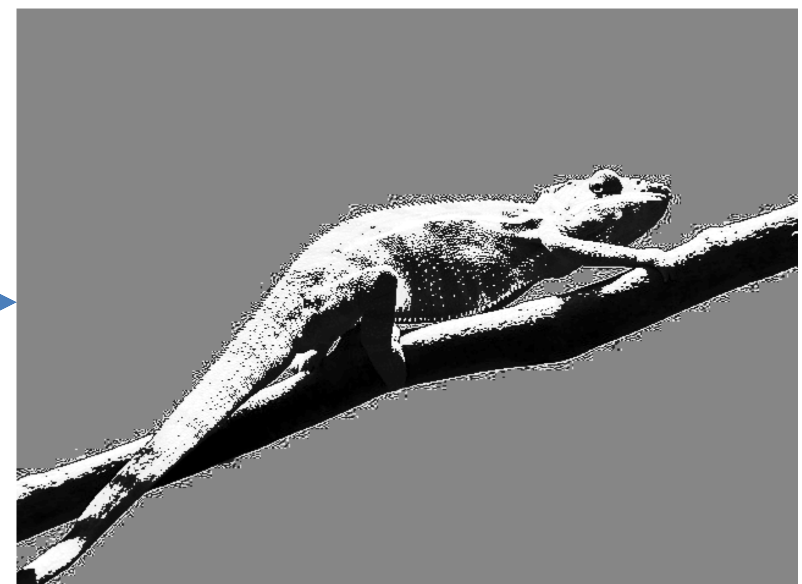
„histogram equalization"

Filter - Punktweise

Annahme der *histogram equalization*: Man weiß, dass das Histogramm des wahren Bildes eine bestimmte Form hat (z.B. ungefähr gleichverteilt ist).

Annahme ok

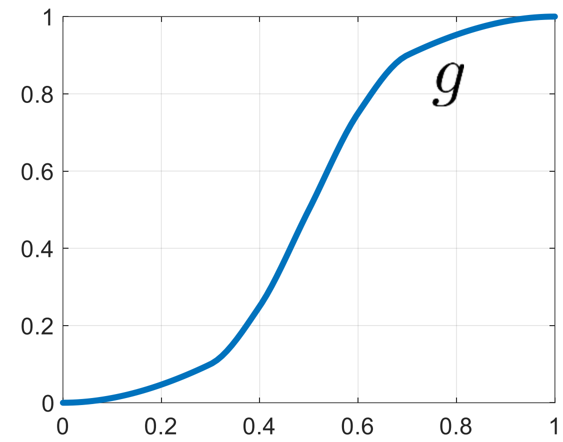
Annahme nicht ok



Lokale lineare Filter

Bislang: Jeder Pixel wird einzeln verändert – der Wert der Nachbarpixel spielt keine Rolle!

(Punktweiser Filter oder pixelweiser Filter)



$$f_{ij}^{new} = g(f_{ij})$$

Nächster Schritt: ***Lokale Filter***

Rechenoperationen zwischen Pixeln und ihren Nachbarn!

Welche Arten von Rechenoperationen kann man durchführen und wie wirken diese sich auf die Bilder aus?

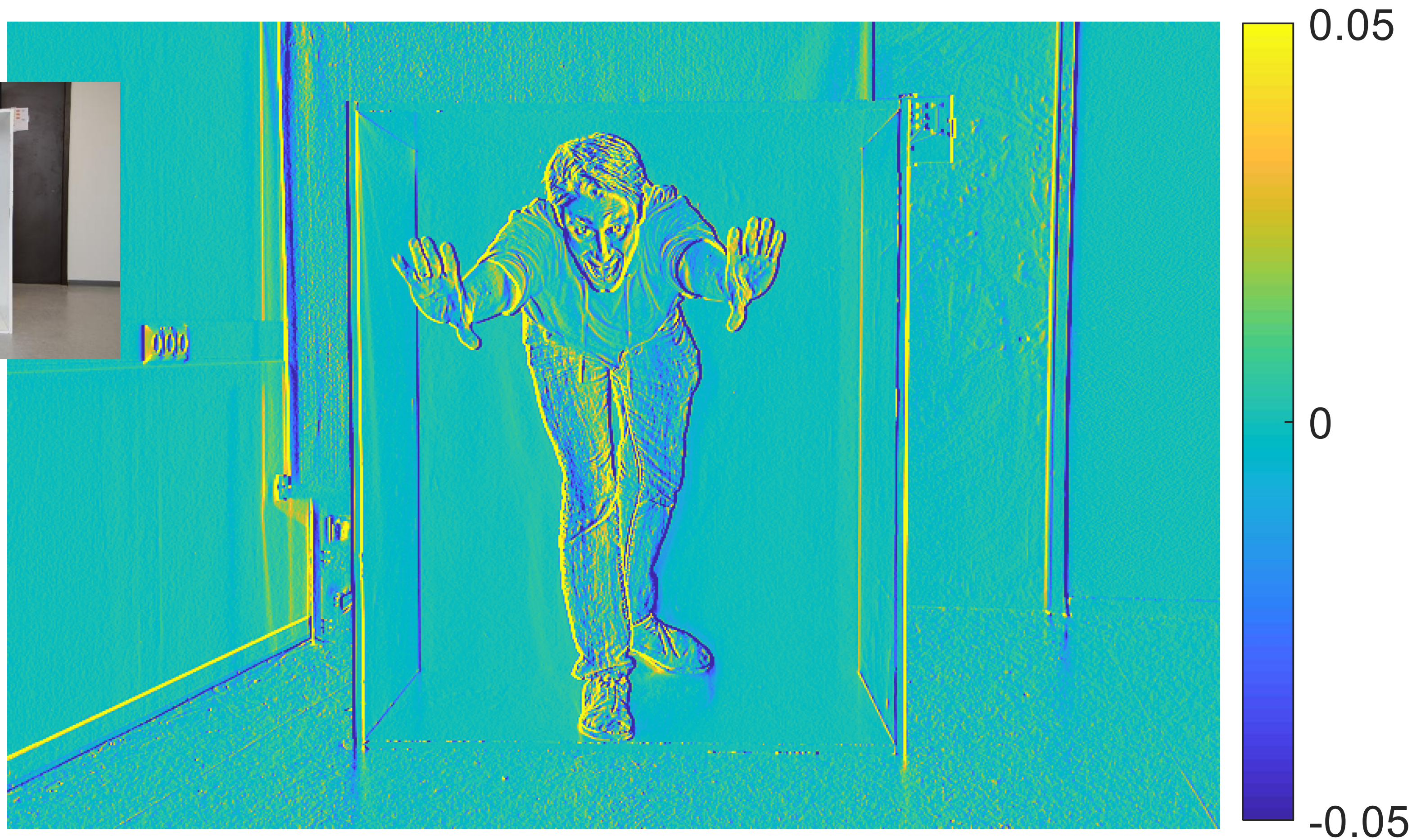
Lokale lineare Filter

Beispiel: Was passiert, wenn man jeweils einen Pixel von seinem Nachbarn subtrahiert?

$$g(f_{i,j}) = f_{i,j} - f_{i+1,j}$$

Lokale lineare Filter

Beispiel: Was passiert, wenn man jeweils einen Pixel von seinem Nachbarn subtrahiert?

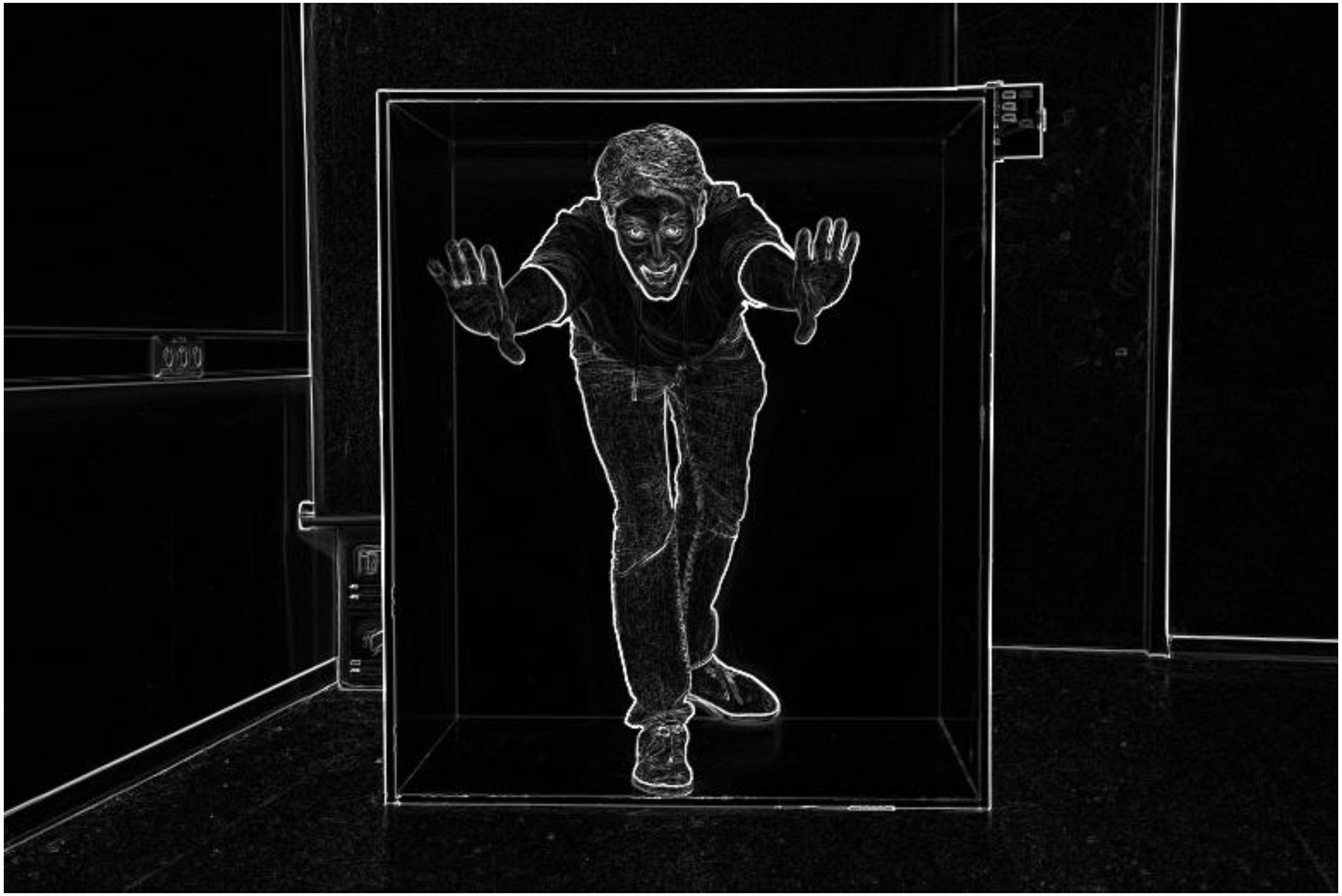


$$g(f_{i,j}) = f_{i,j} - f_{i,j+1}$$

Michael Möller – michael.moeller@uni-siegen.de

Lokale lineare Filter

Anwendung 1 (Details später): Kanten detektieren!



$$g(f_{i,j}) = |f_{i,j} - f_{i,j+1}| + |f_{i,j} - f_{i+1,j}|$$

Lokale lineare Filter

Anwendung 2: Schärfen

$$g(f_{i,j}) = f_{i,j} - f_{i,j+1}$$ >0 falls Pixel größer als sein rechter Nachbar

$$g(f_{i,j}) = f_{i,j} - f_{i,j-1}$$ >0 falls Pixel größer als sein linker Nachbar

$$g(f_{i,j}) = f_{i,j} - f_{i+1,j}$$ >0 falls Pixel größer als sein unterer Nachbar

$$g(f_{i,j}) = f_{i,j} - f_{i-1,j}$$ >0 falls Pixel größer als sein oberer Nachbar

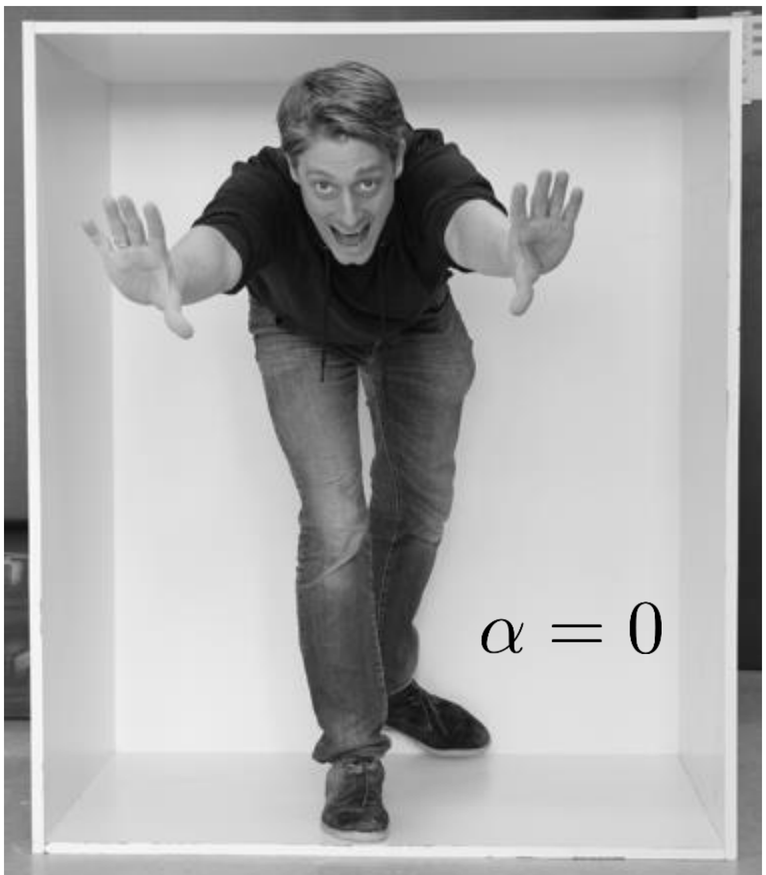
$$g(f_{i,j}) = \frac{1}{4}\Big(\big(f_{i,j} - f_{i-1,j}\big) + \big(f_{i,j} - f_{i+1,j}\big) + \big(f_{i,j} - f_{i,j-1}\big) + \big(f_{i,j} - f_{i,j+1}\big)\Big)$$

>0 falls Pixel im Durchschnitt größer als seine Nachbarn

Idee: Addiere obige Funktion zum aktuellen Pixelwert hinzu um Unterschiede zu verstärken und das Bild so zu schärfen!

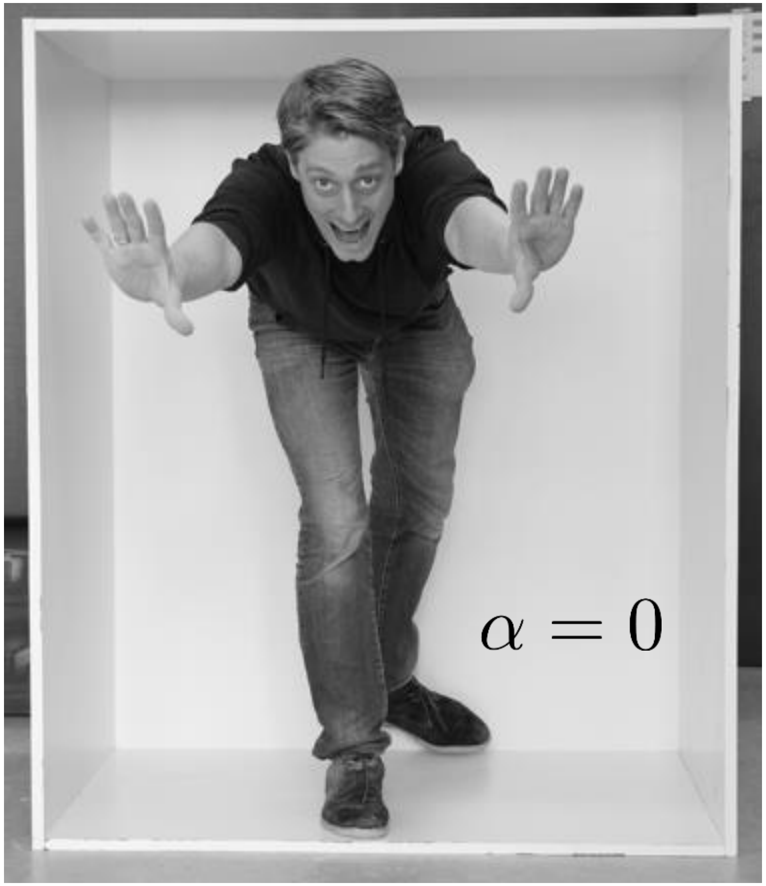
Lokale lineare Filter

$$g(f_{i,j}) = f_{i,j} + \frac{\alpha}{4}\Big(\big(f_{i,j} - f_{i-1,j}\big) + \big(f_{i,j} - f_{i+1,j}\big) + \big(f_{i,j} - f_{i,j-1}\big) + \big(f_{i,j} - f_{i,j+1}\big)\Big)$$



Lokale lineare Filter

$$g(f_{i,j}) = f_{i,j} + \frac{\alpha}{4}\Big(\big(f_{i,j} - f_{i-1,j}\big) + \big(f_{i,j} - f_{i+1,j}\big) + \big(f_{i,j} - f_{i,j-1}\big) + \big(f_{i,j} - f_{i,j+1}\big)\Big)$$



Schärfungsfilter

Schärfungsfilter addieren zum aktuellen Pixelwert eine gewichtete Linearkombination der Nachbarpixel derart, dass Unterschiede verstärkt werden. Ein typisches Beispiel eines solchen Filters ist

$$g(f_{i,j}) = \left(1 + \frac{\alpha}{4}\right) f_{i,j} - \frac{\alpha}{4}\left(f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1}\right)$$

$$g(f_{i,j}) = \text{Summe}\left(\begin{array}{|c|c|c|} \hline & f_{i-1,j} & \\ \hline f_{i,j-1} & f_{i,j} & f_{i,j+1} \\ \hline & f_{i+1,j} & \\ \hline \end{array} \;\underset{\text{multiplizieren}}{\overset{\text{Punktweise}}{\bullet}}\; \begin{array}{|c|c|c|} \hline & -\frac{\alpha}{4} & \\ \hline -\frac{\alpha}{4} & 1 + \frac{\alpha}{4} & -\frac{\alpha}{4} \\ \hline & -\frac{\alpha}{4} & \\ \hline \end{array} \right)$$

Korrelationsfilter

$$g(f_{i,j}) = \sum \left(\begin{array}{|c|c|c|} \hline f_{i-1,j-1} & f_{i-1,j} & f_{i-1,j+1} \\ \hline f_{i,j-1} & f_{i,j} & f_{i,j+1} \\ \hline f_{i+1,j-1} & f_{i+1,j} & f_{i+1,j+1} \\ \hline \end{array} \overset{\text{Punktw. Multi.}}{\bullet} \begin{array}{|c|c|c|} \hline k_{i-1,j-1} & k_{i-1,j} & k_{i-1,j+1} \\ \hline k_{i,j-1} & k_{i,j} & k_{i,j+1} \\ \hline k_{i+1,j-1} & k_{i+1,j} & k_{i+1,j+1} \\ \hline \end{array} \right)$$

Filter dieser Art heißen ***Korrelationsfilter***.
Die Größe der Betrachteten Umgebung darf beliebig variieren (hier dargestellt: 3x3 Umgebung). Formal schreiben wir

$$(k \star f)_{i,j} = \sum_{k=-w_y}^{w_y} \sum_{l=-w_x}^{w_x} f_{i+k,j+l} \;\; k_{w_y+k,w_x+l}$$

Und bezeichnen dies als Korrelationsfilter der Größe $(2w_y+1) \times (2w_x+1)$
Das k bezeichnet man als ***Kern (kernel)***.

Korrelationsfilter am Rand

Die Formel

$$(k \star f)_{i,j} = \sum_{k=-w_y}^{w_y} \sum_{l=-w_x}^{w_x} f_{i+k,j+l} \;\; k_{w_y+k,w_x+l}$$

Ist unvollständig, da sie die Fälle $i+k<0$, $j+l<0$, $i+k>0$ und $j+l>n_x$ ignoriert (und die obige Formel dort nicht ausrechenbar ist).

Für die Randbehandlung gibt es unterschiedliche Möglichkeiten:

Zero-padding: $f_{s,t} = 0 \quad \text{falls } s<0 \text{ oder } t<0 \text{ oder } s>n_y \text{ oder } t>n_x$

Replicate:

$f_{s,t} = f_{0,t} \quad \text{falls } s<0$ $\qquad$ $f_{s,t} = f_{s,0} \quad \text{falls } t<0$

$f_{s,t} = f_{n_y,t} \quad \text{falls } s>n_y$ $\qquad$ $f_{s,t} = f_{s,n_x} \quad \text{falls } t>n_x$

Cyclic: $f_{s,t} = f_{\text{mod}(s,n_y),\text{mod}(t,n_x)}$

Faltungen

Wir nennen Filter der Form

$$(k * f)_{i,j} = \sum_{k=-w_y}^{w_y} \sum_{l=-w_x}^{w_x} f_{i-k,j-l} \;\; k_{w_y+k,w_x+l}$$

(mit Randbedingungen) ***Faltungsfilter***. Diese sind das gleiche wie Korrelationsfilter mit einem in x- und y-Richtung gespiegelten Kern k.

Warnung:
Die Begriffe werden in der Literatur nicht immer ganz trennscharf verwendet! Insbesondere im Bereich des Deep Learnings sind Convolutional Layers (deutsch: Faltungsschichten) eigentlich Korrelationsfilter. Die Korrelationsfilter nennt man auch ***Kreuz-Korrelation***.

Faltungen

Was kann man mit Faltungen/Kreuz-Korrelationen alles tun?

Schärfen, z.B.

$$k = \begin{pmatrix} 0 & -\alpha/4 & 0 \\ -\alpha/4 & 1+\alpha & -\alpha/4 \\ 0 & -\alpha/4 & 0 \end{pmatrix}$$

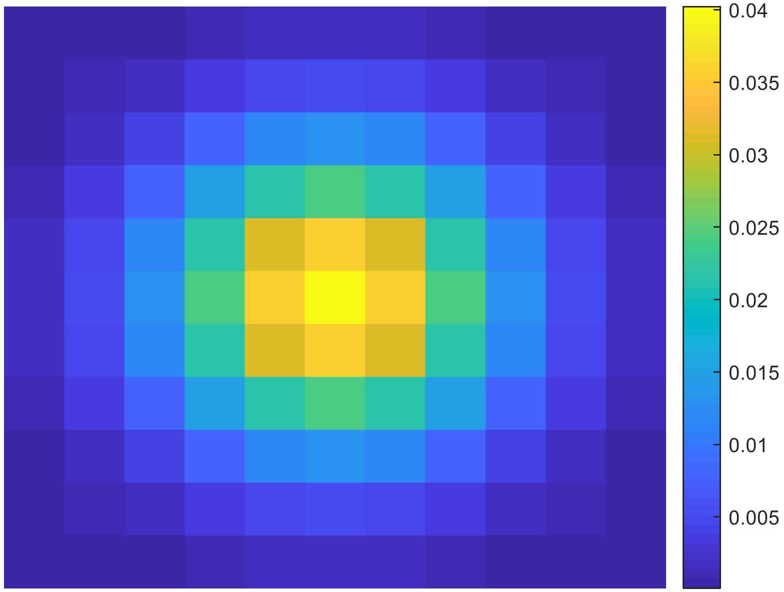
$$k = \begin{pmatrix} -\alpha/4 & -\alpha/2 & -\alpha/4 \\ -\alpha/2 & 1+3\alpha & -\alpha/2 \\ -\alpha/4 & -\alpha/2 & -\alpha/4 \end{pmatrix}$$

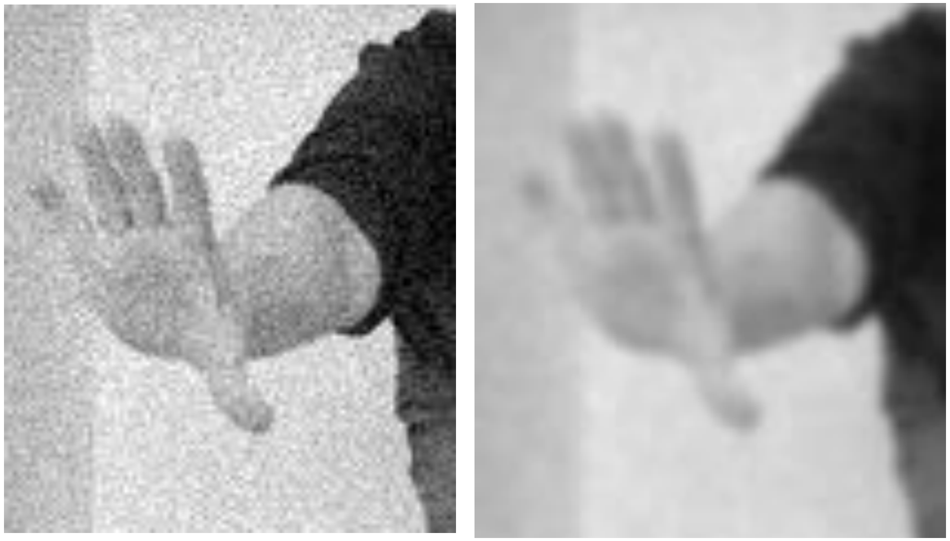
Faltungen

Was kann man mit Faltungen/Kreuz-Korrelationen alles tun?

Glätten, z.B.

$$\tilde{k}_{ij} = \exp\left(-\frac{(i-i_c)^2+(j-j_c)^2}{\sigma^2}\right), \qquad k_{ij} = \frac{1}{\sum_{r,s}\tilde{k}_{r,s}}\tilde{k}_{i,j}$$



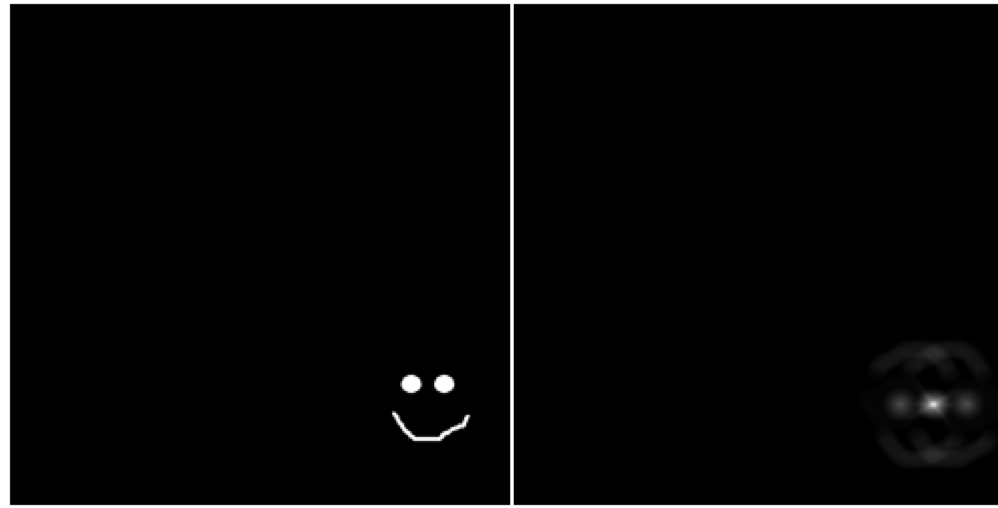


Faltungen

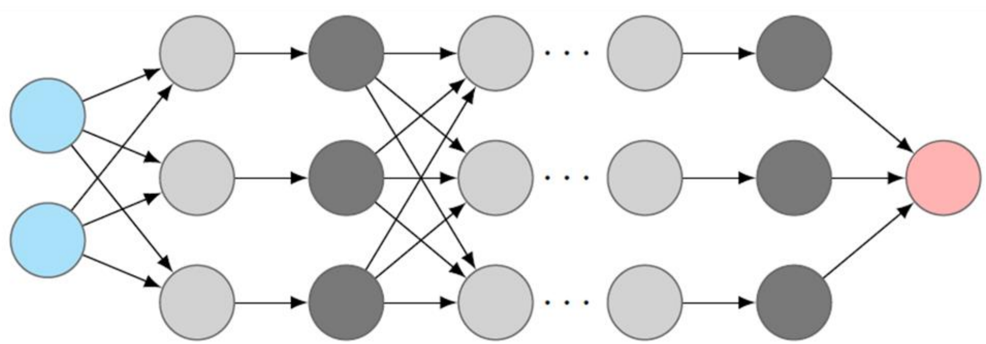
Was kann man mit Faltungen/Kreuz-Korrelationen alles tun?

Strukturen finden

$k =$ 



Neuronale Netzwerke zusammensetzen



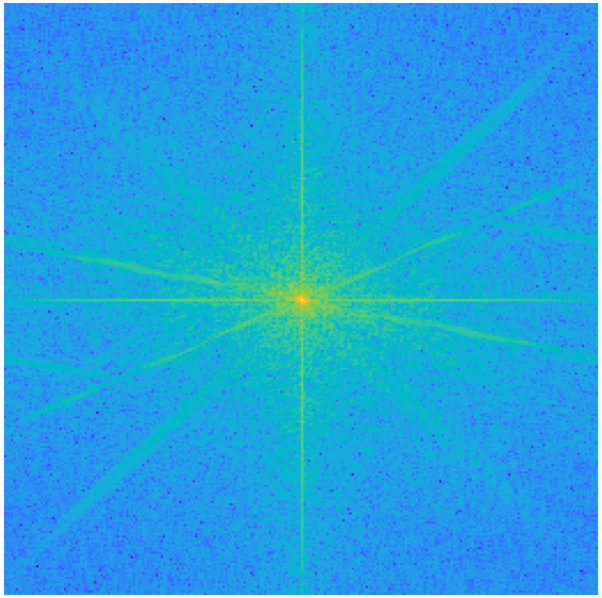
Ist eine Giraffe auf dem Bild zu sehen?

Faltungen

Wie kann man Faltungen noch interpretieren?

Antwort in der nächsten Vorlesung:
Bilder sind eine Überlagerung unterschiedlicher Frequenzen. Faltungen dämpfen oder verstärken bestimmte Frequenzen!

Ortsdarstellung



Frequenzdarstellung
(Fourier Koeffizienten)