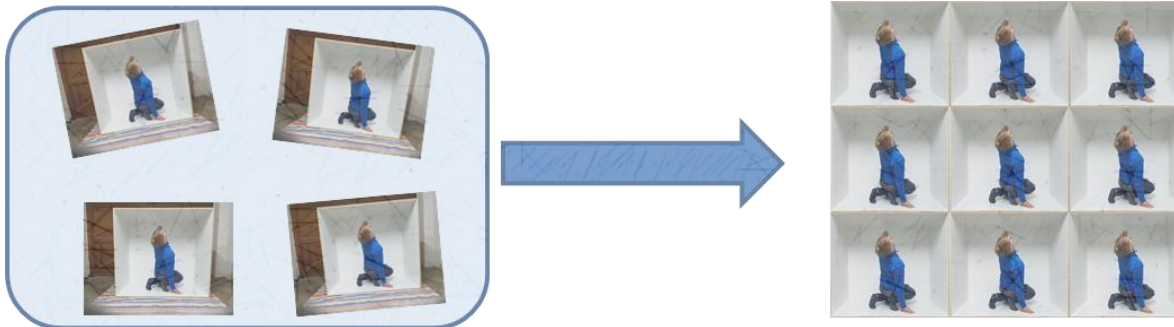


Digitale Bildverarbeitung 1

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



Ziel: Aufteilen des Bildes in „sinnvolle“ Bereiche

- Was sinnvoll ist, hängt von der Anwendung ab

Aktuelle, (aber schwierige) Anwendung: Segmentierung für autonomes Fahren

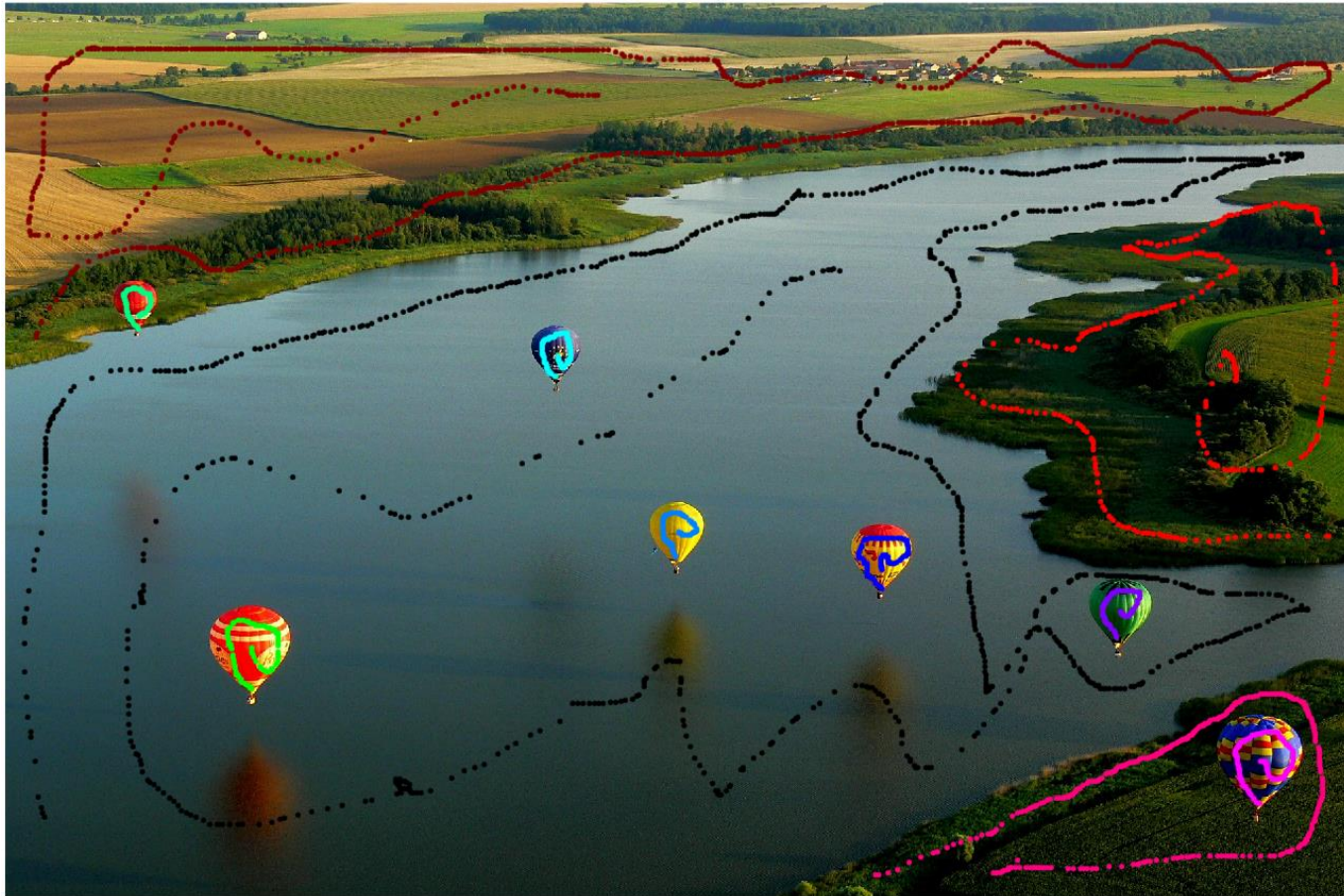


<https://www.cityscapes-dataset.com/dataset-overview/>

Ziel: Aufteilen des Bildes in „sinnvolle“ Bereiche

- Was sinnvoll ist, hängt von der Anwendung ab

Weiteres Beispiel: Segmentierung in Photobearbeitung, z.B. mit „Scribbles“



Ziel: Aufteilen des Bildes in „sinnvolle“ Bereiche

- Was sinnvoll ist, hängt von der Anwendung ab

Weiteres Beispiel: Segmentierung in Photobearbeitung, z.B. mit „Scribbles“



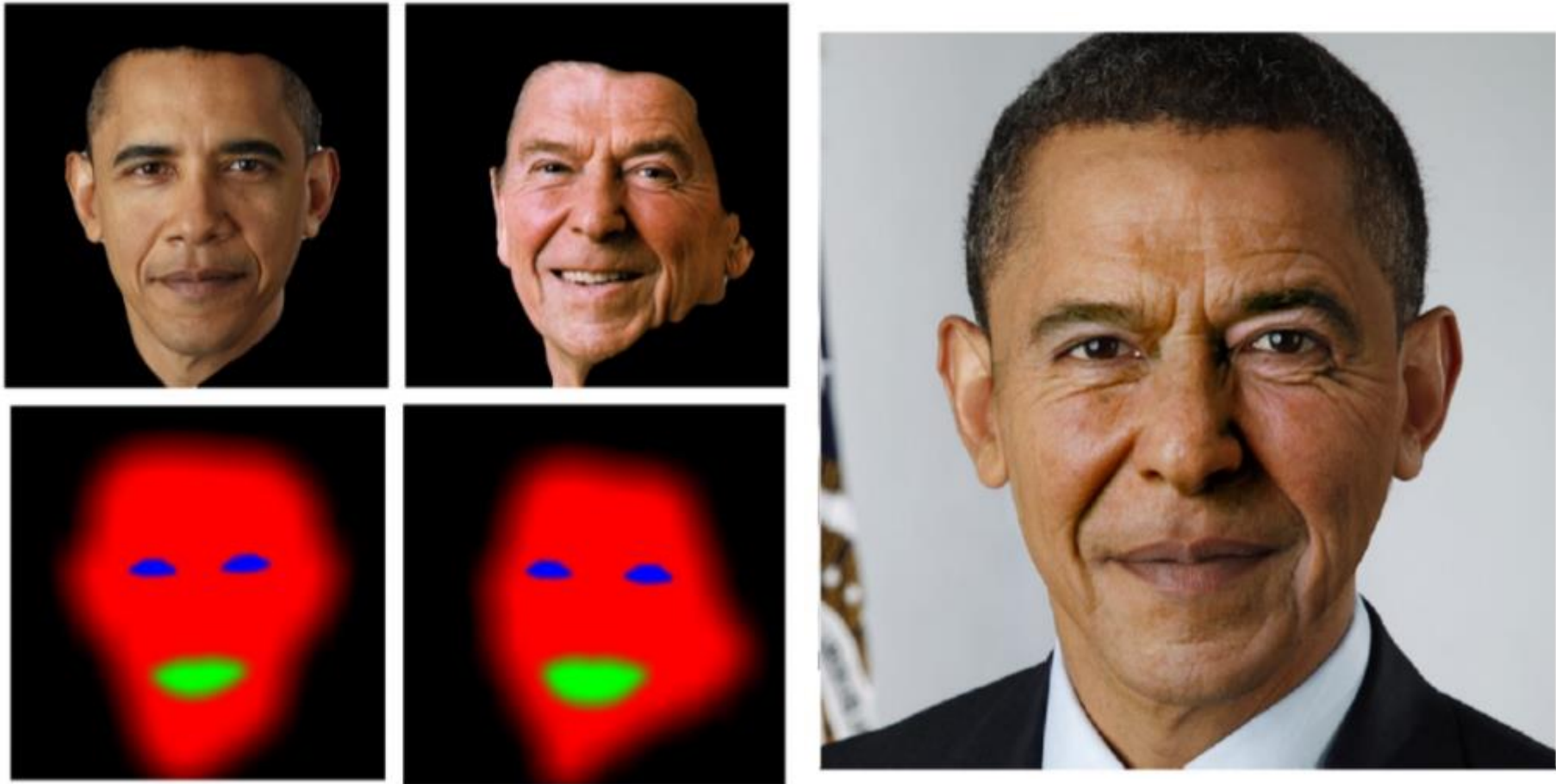
Ziel: Aufteilen des Bildes in „sinnvolle“ Bereiche

- Was sinnvoll ist, hängt von der Anwendung ab

Weiteres Beispiel: Segmentierung in Photobearbeitung, z.B. mit „Scribbles“



Weiteres Beispiel: Segmentierung in Photobearbeitung, z.B. mit „Scribbles“



Martin Benning, Michael Moeller, Raz Nosssek, Martin Burger, Daniel Cremers, Guy Gilboa, Carola Schönlieb, *Nonlinear Spectral Image Fusion*. SSVM 2017.

Und damit kann man dann viel Unsinn machen... 😊



Martin Benning, Michael Moeller, Raz Nosssek, Martin Burger, Daniel Cremers, Guy Gilboa, Carola Schönlieb, *Nonlinear Spectral Image Fusion*. SSVM 2017.

Segmentierung, ohne zu wissen was genau man segmentieren möchte: Clustering

Eingabebild



Segmentierung in 6 Klassen



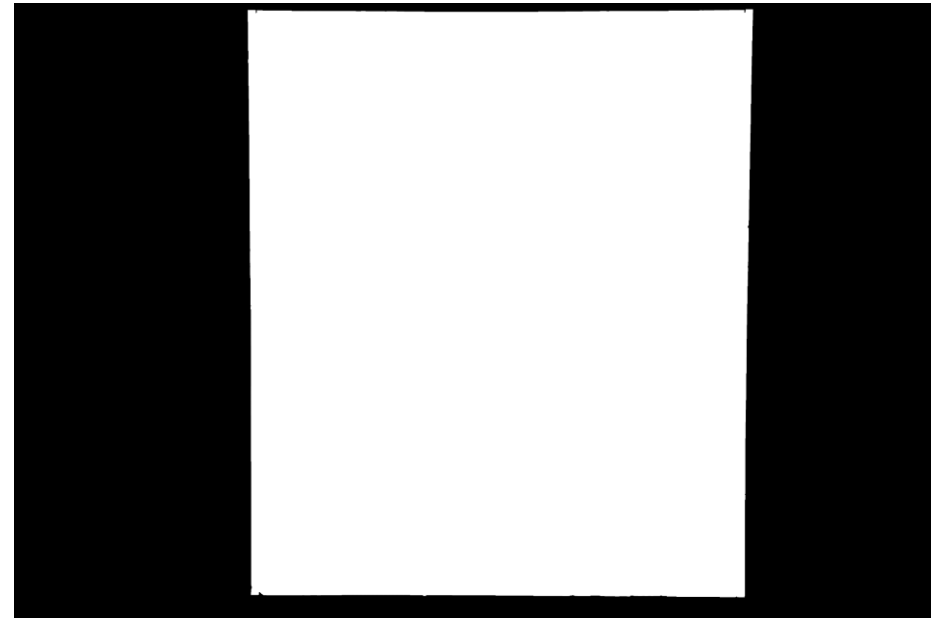
Bilder von <https://towardsdatascience.com/introduction-to-image-segmentation-with-k-means-clustering-83fd0a9e2fc3>

Unser Ziel heute: 2 Regionensegmentierung für unsere Kistenbilder!

Eingabebild



Segmentierung



Unser Ziel heute: 2 Regionensegmentierung für unsere Kistenbilder!



Unser Ziel heute: 2 Regionensegmentierung für unsere Kistenbilder!

Erste Frage: Wie repräsentiere ich eigentlich eine *Segmentierung*?

Für 2 Regionen: Eine Segmentierung des Farbbildes $f \in \mathbb{R}^{n_y \times n_x \times 3}$ ist ein Bild $g \in \mathbb{R}^{n_y \times n_x}$ mit der gleichen Anzahl an Pixeln für welches gilt

$$g_{i,j} = \begin{cases} 1 & \text{falls der Pixel } (i,j) \text{ zum Vordergrund gehört} \\ 0 & \text{falls der Pixel } (i,j) \text{ zum Hintergrund gehört} \end{cases}$$

Was sind typischerweise Verfahren um eine Segmentierung zu erhalten?

- **Modellbasiert:** Man überlegt sich Kriterien welche definieren, wie sich das gesuchte Objekt vom Hintergrund unterscheidet
- **Lernbasiert:** Man stellt einem maschinellen Lernverfahren viele Beispiele von Bildern und der gewünschten Segmentierung zur Verfügung, aus denen eine Rechenvorschrift für die Segmentierung gelernt wird.

Wir beschäftigen uns hier nur mit modellbasierten Verfahren. Die Vorlesungen *statistische Lerntheorie*, *Deep Learning* und auch das *Praktikum Digitale Bildverarbeitung* geben weitere Einblicke in das zweite Themenfeld.

Modellbasierte Segmentierung: Schritt 1 – was unterscheidet den Vordergrund vom Hintergrund?



Erste Idee: Die Helligkeit

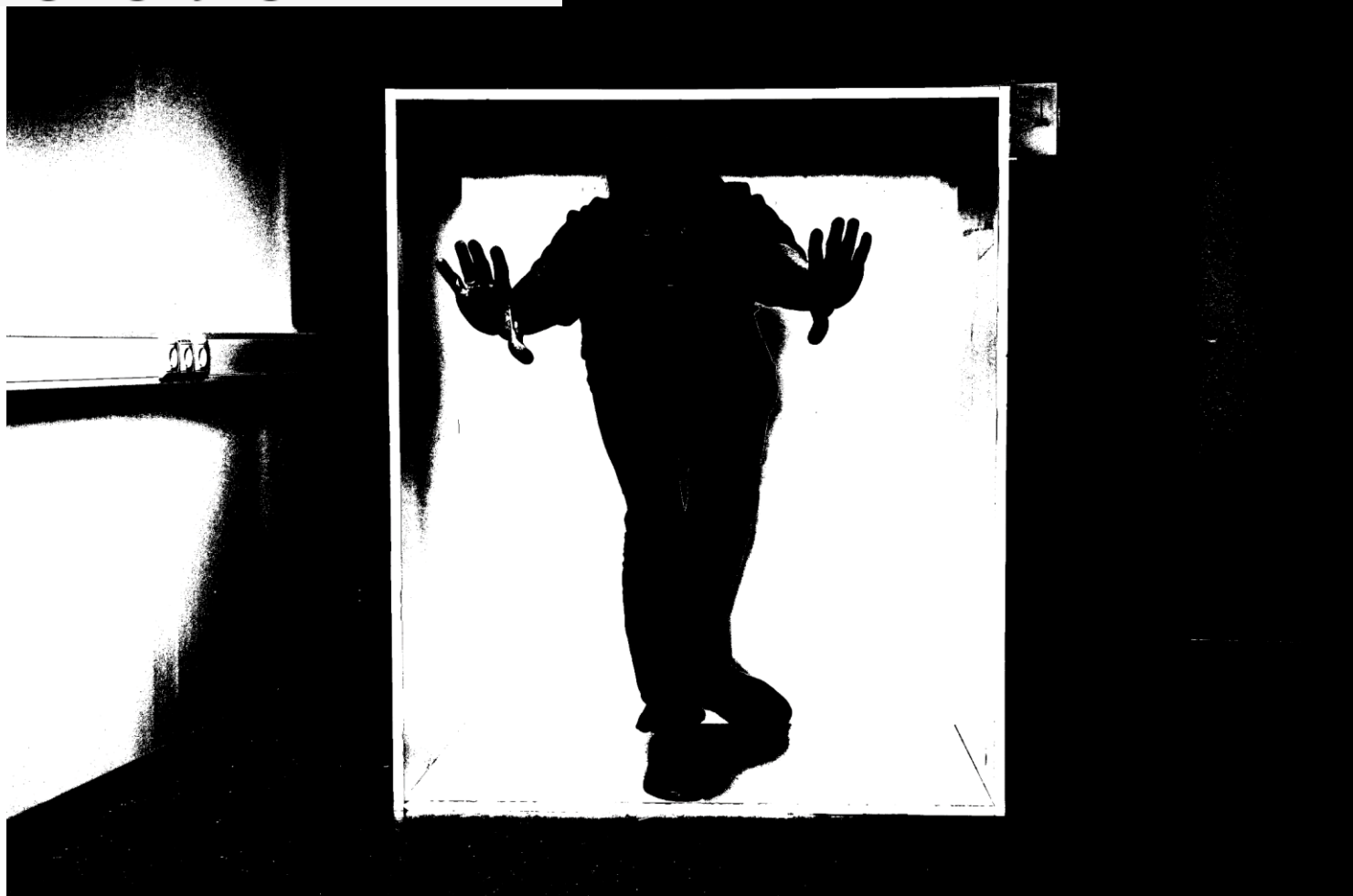
```
greyImg = (testing[:, :, 0] + testing[:, :, 1] + testing[:, :, 2]) / 3
```



Erste Idee: Die Helligkeit „thresholden“ (einen Schwellwert setzen)

```
threshold = 0.75
```

```
binaryImg = greyImg > threshold
```



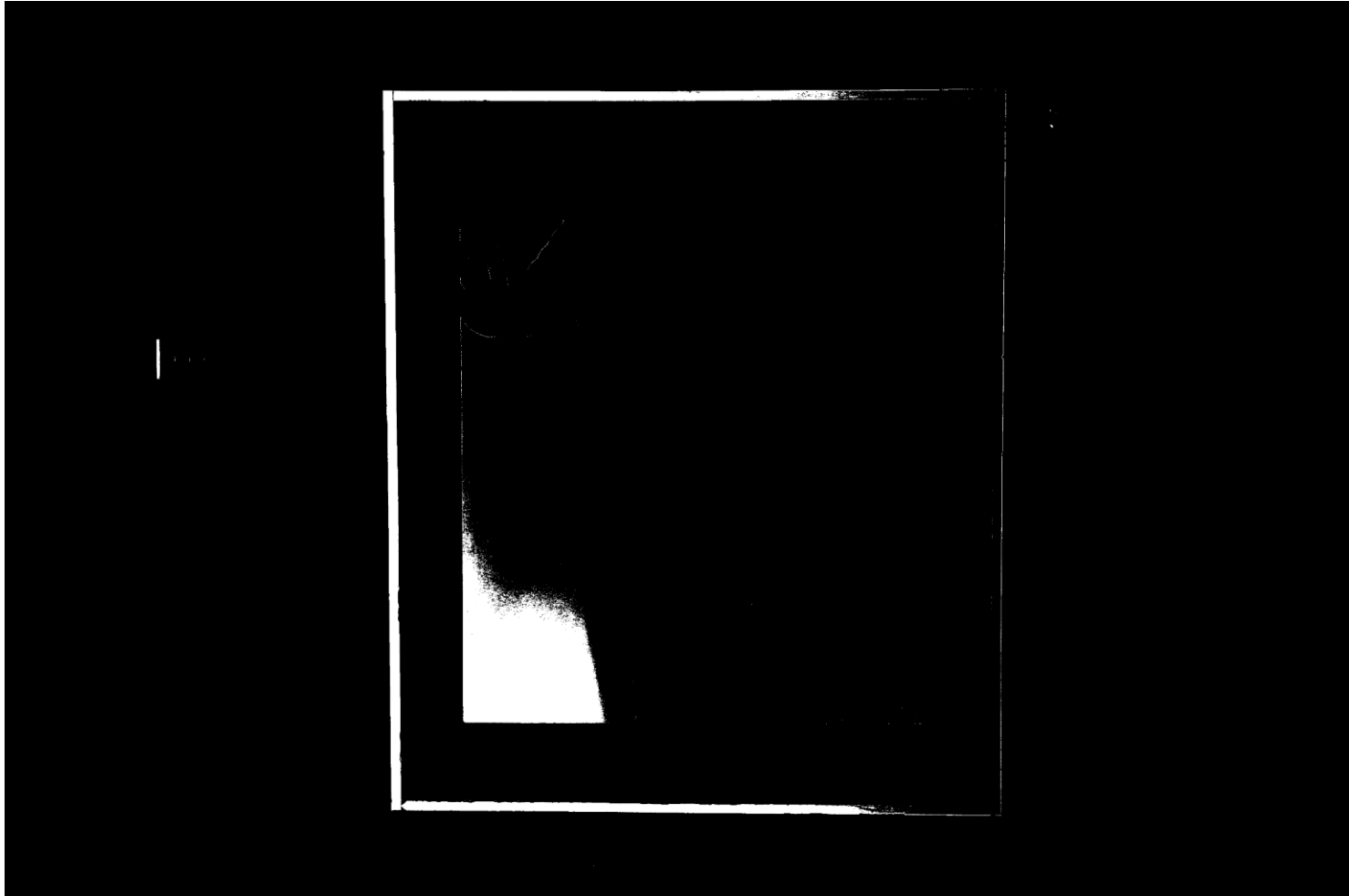
Erste Idee: Die Helligkeit „thresholden“ (einen Schwellwert setzen)

Threshold zu niedrig (hier 0.65):



Erste Idee: Die Helligkeit „thresholden“ (einen Schwellwert setzen)

Threshold zu hoch (hier 0.9):



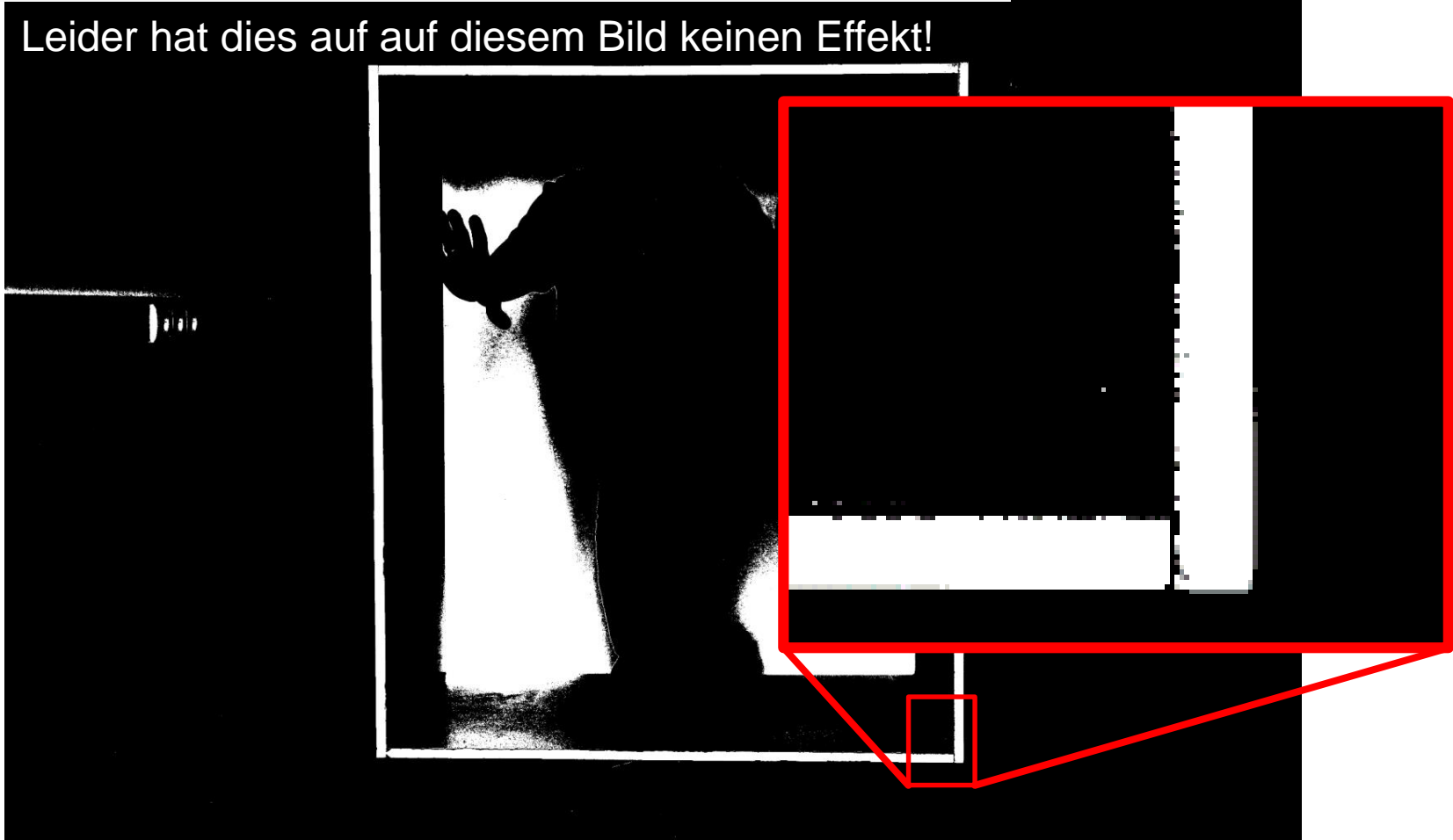
Selbst mit einem ordentlichen Threshold sind wir weit von einer guten Segmentierung weg, was nun?



Es gibt die Möglichkeit Löcher in weißen Regionen auszufüllen

```
import scipy.ndimage.morphology as morph  
binaryImg = morph.binary_fill_holes(binaryImg)
```

Leider hat dies auf auf diesem Bild keinen Effekt!



Wir brauchen also eine Möglichkeit kleine Fehler in der Segmentierung zu korrigieren!

Wir werden nun 2 solche Operationen kennenlernen,
die (morphologische) **Dilatation** und die (morphologische) **Erosion**

Formale Definition:

Für $G \subset \mathbb{Z}^2$ und $S \subset \mathbb{Z}^2$ ist die Dilatation von G mit S gegeben durch

$$G \oplus S := \{x \mid G \cap S_x \neq \emptyset\}$$

wobei

$$S_x := \{x + y \mid y \in S\}$$

die Verschiebung von S nach x beschreibt.

Hier sollen G die Koordinaten der Vordergrundpixel sein und S ein sogenanntes *Strukturelement*

Vielleicht erstmal anschaulich für ein Bild

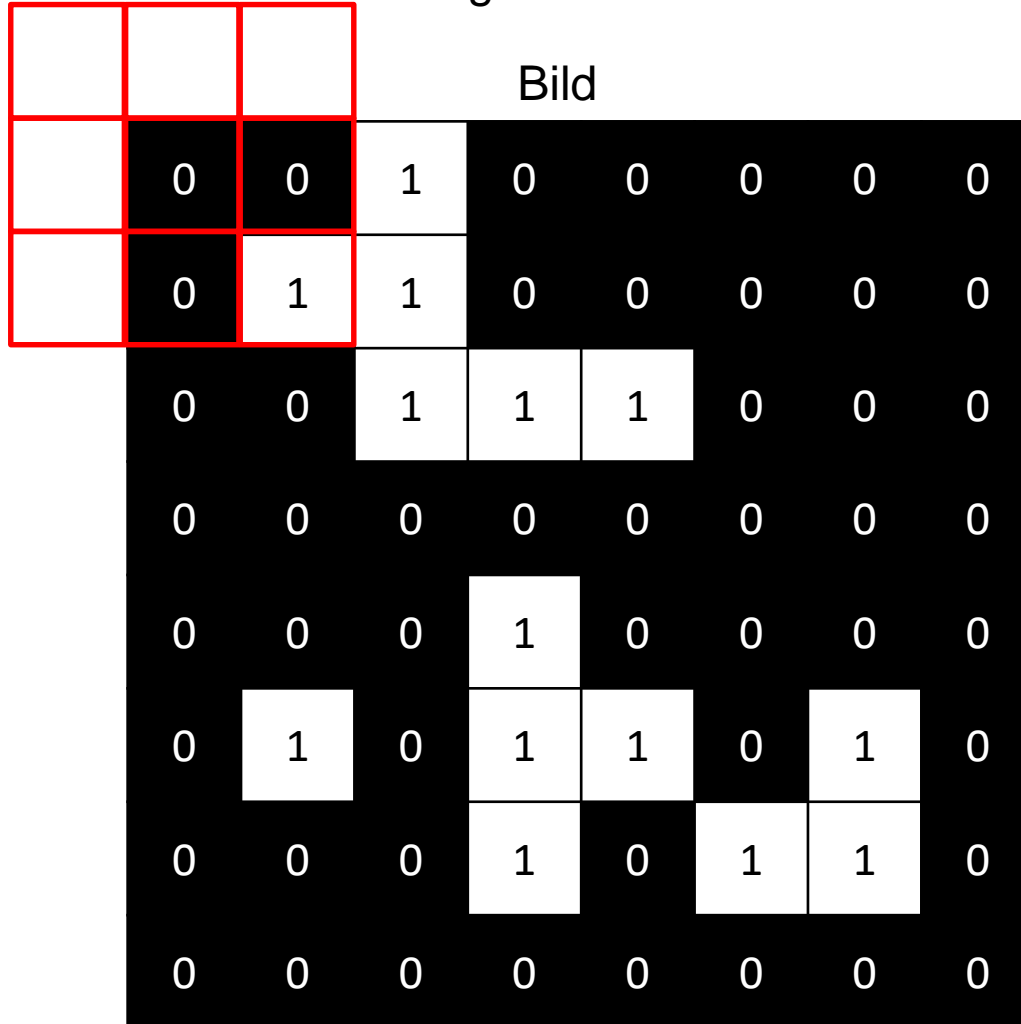
Bild

0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



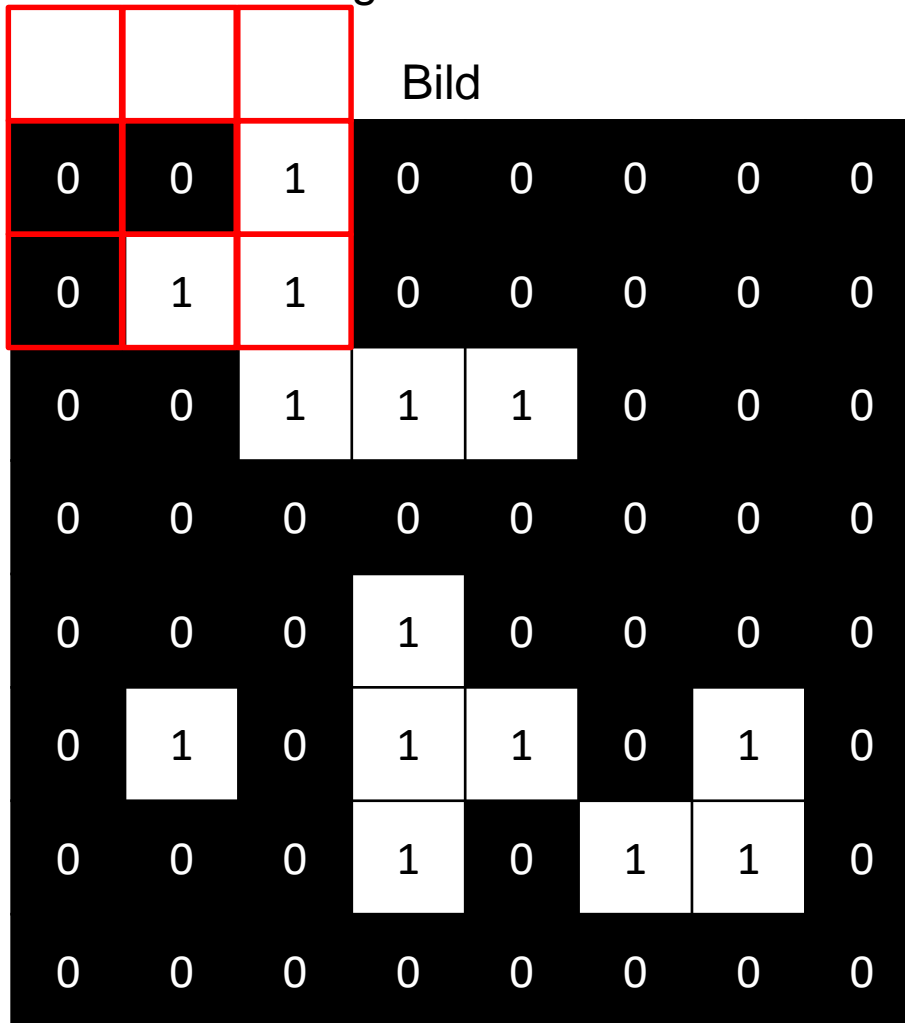
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



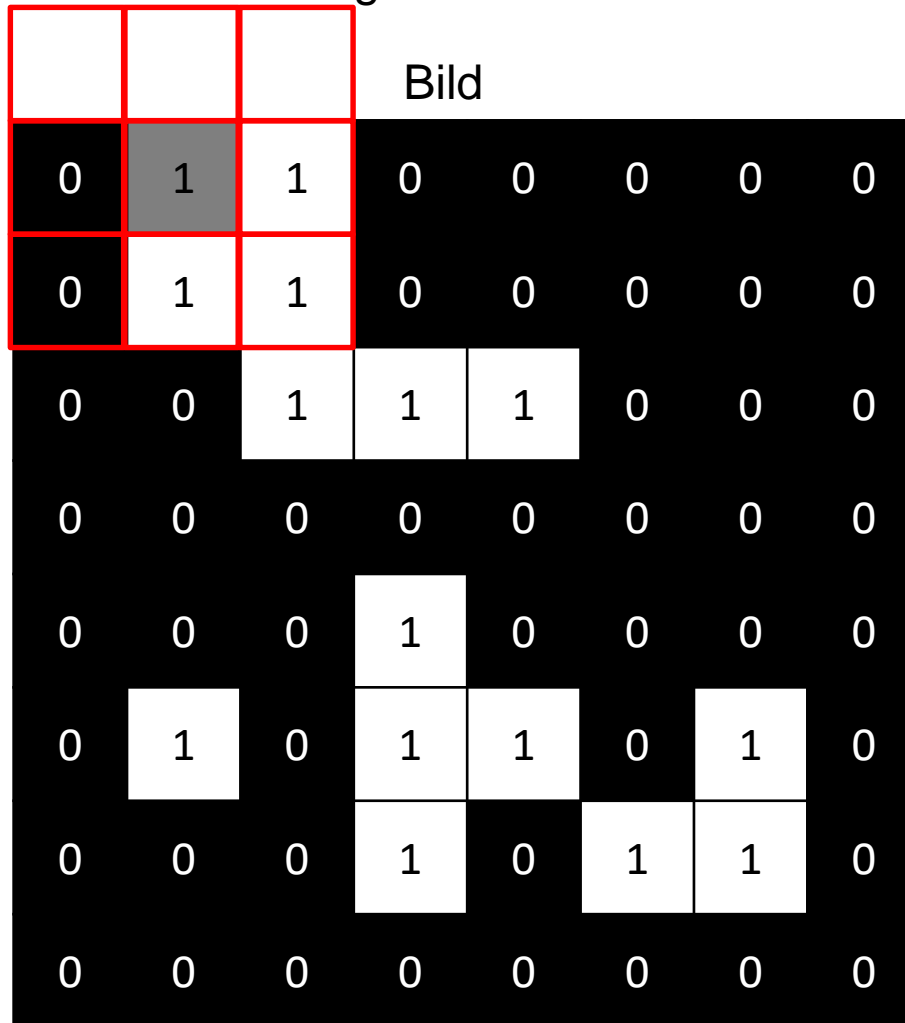
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



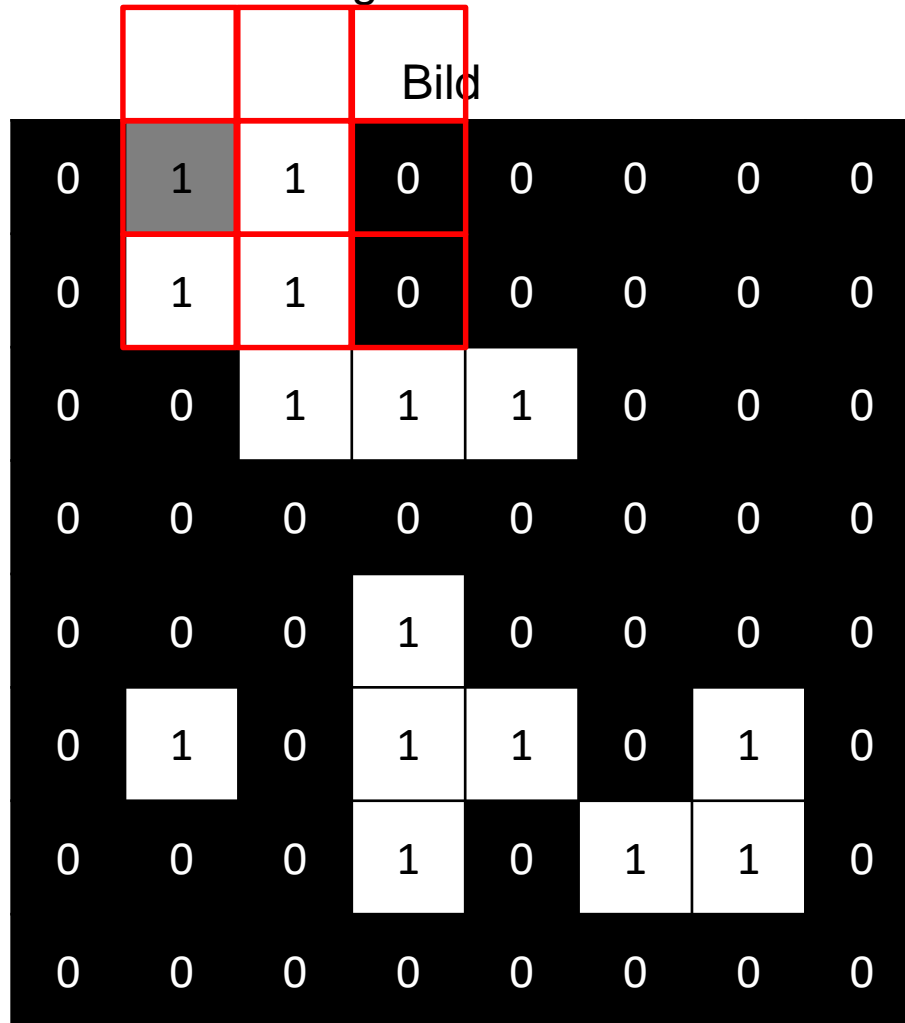
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



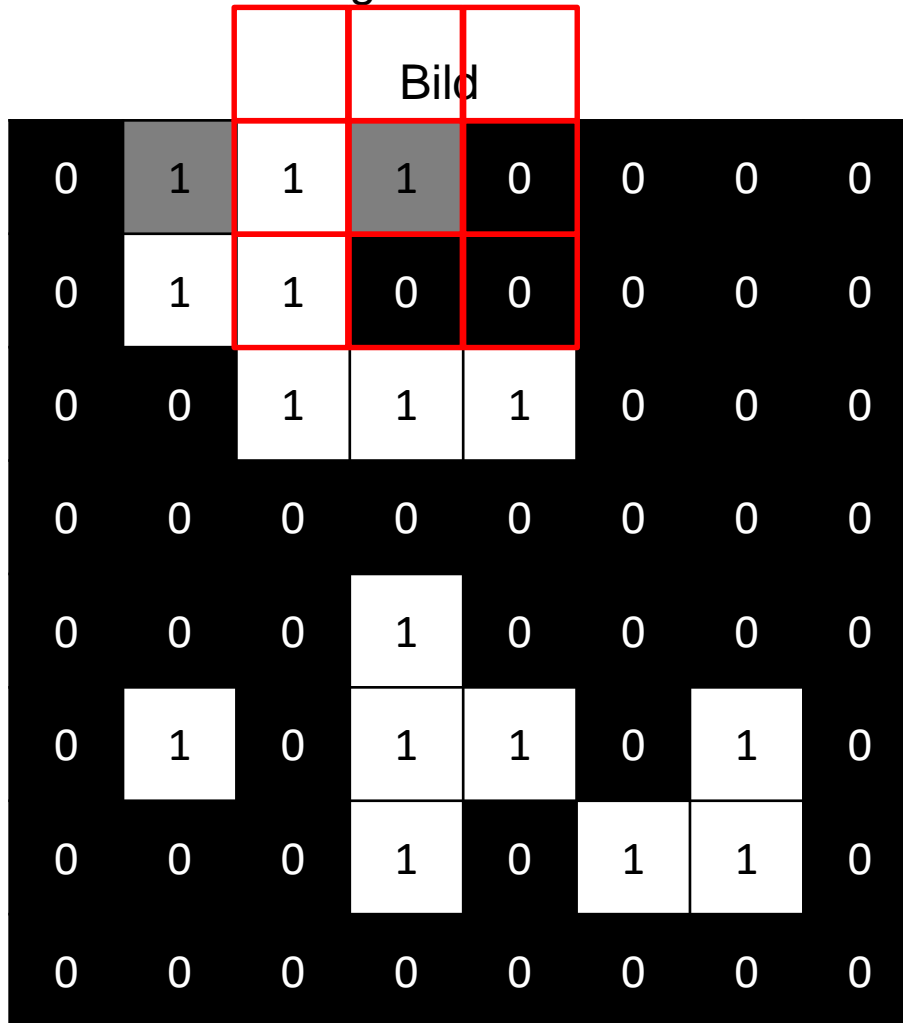
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



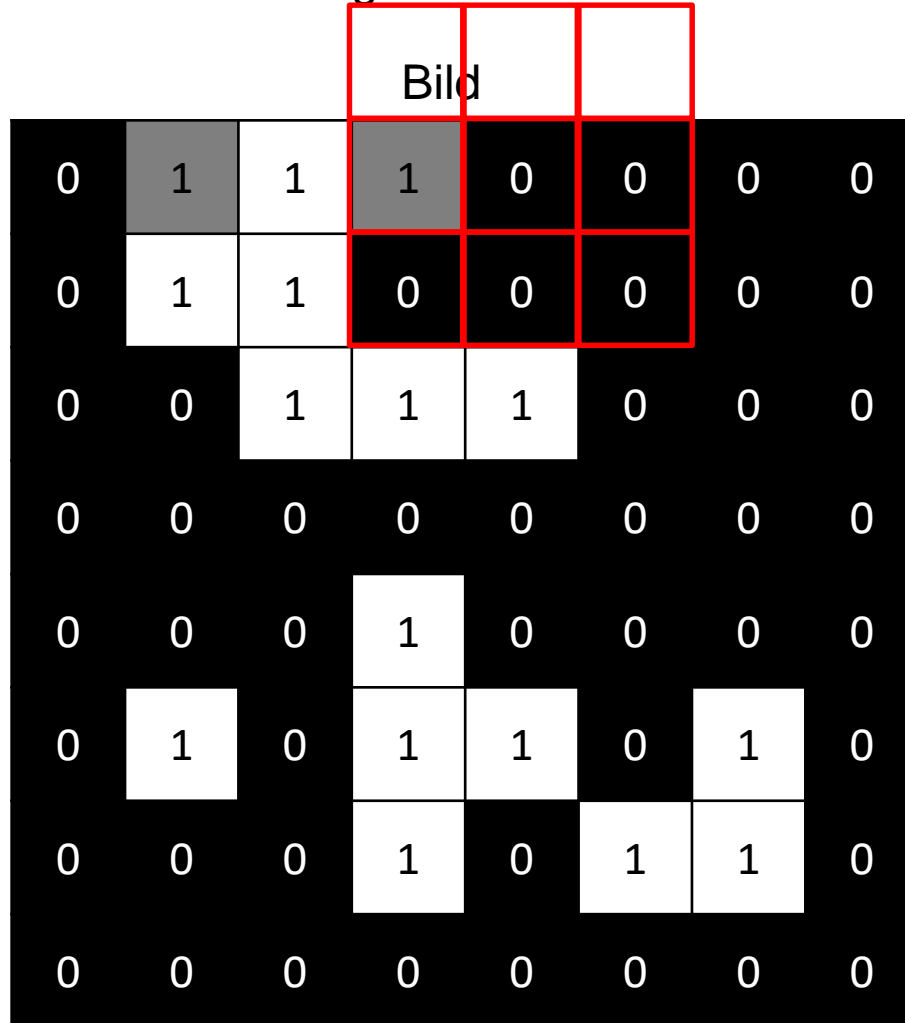
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



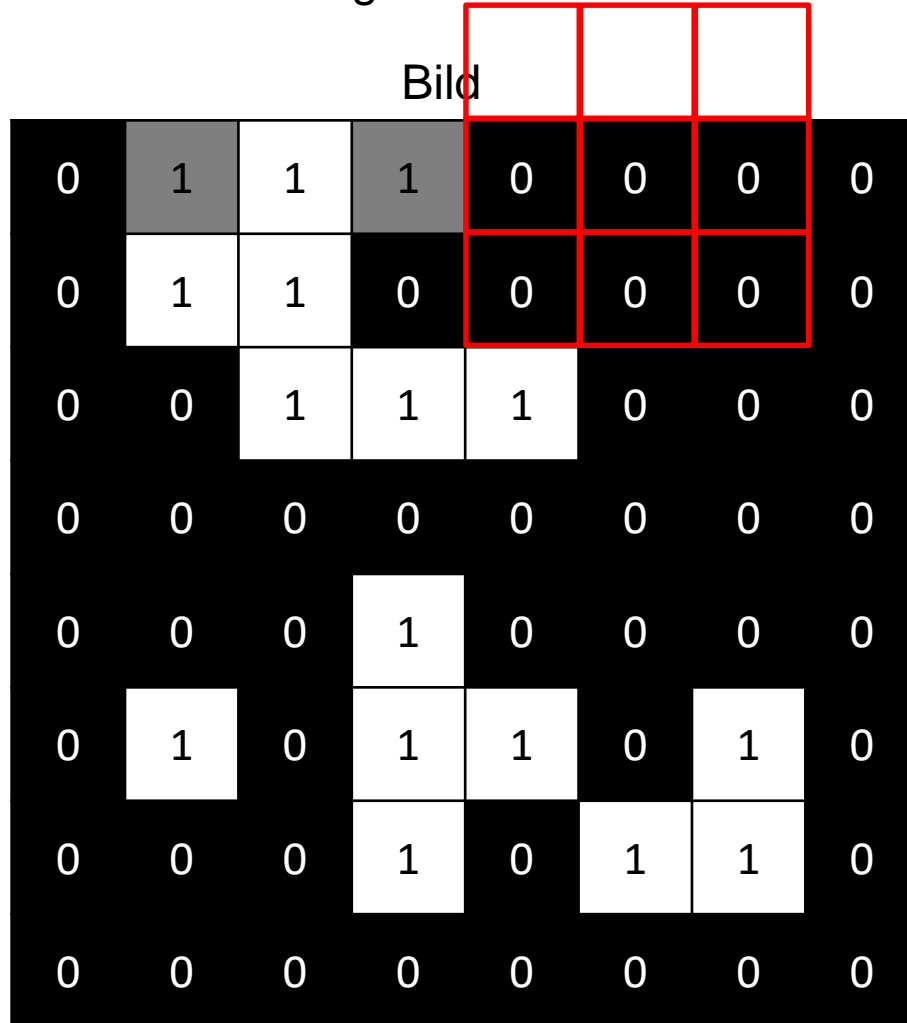
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



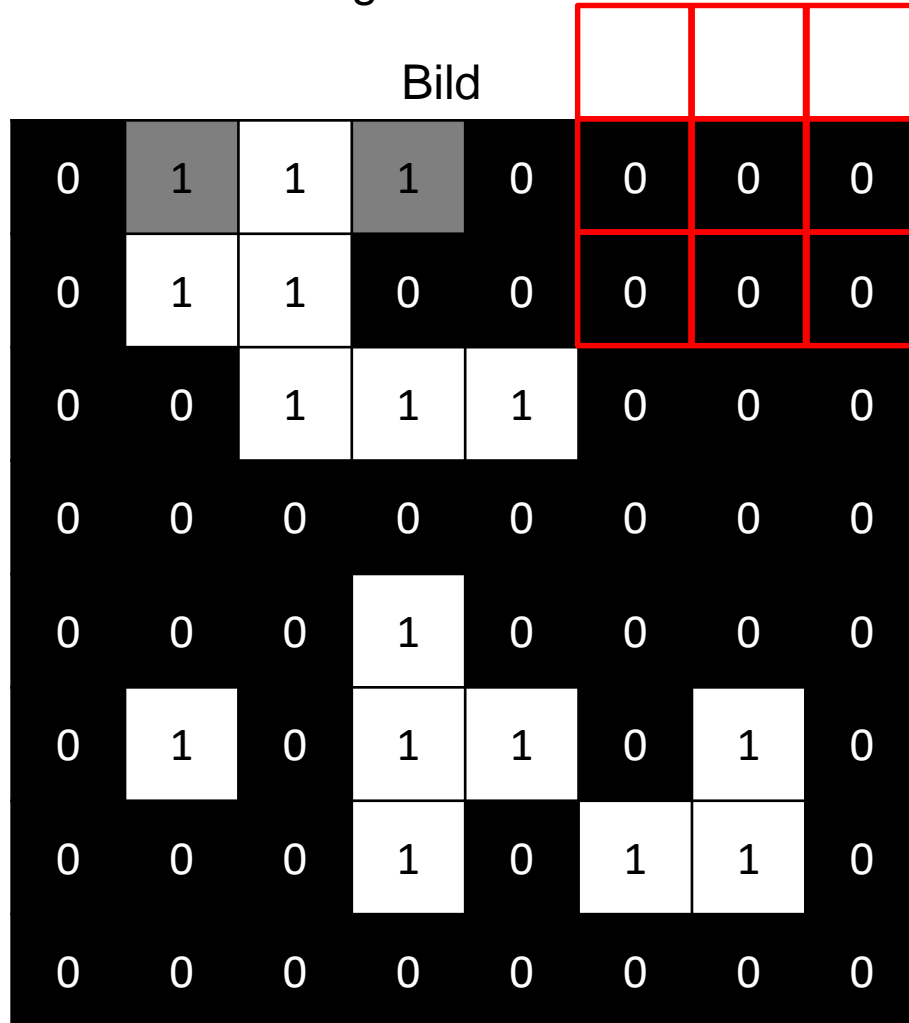
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



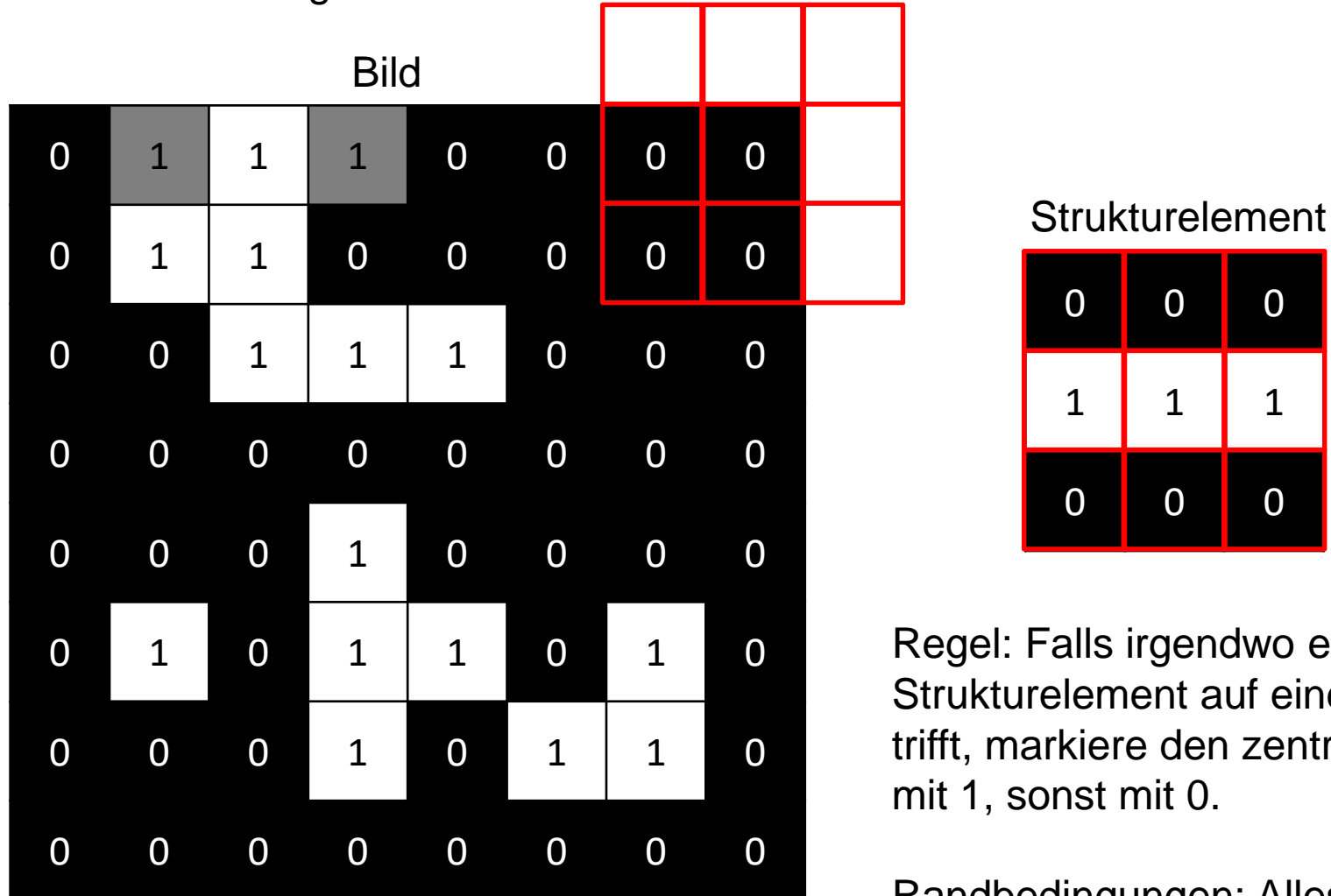
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

	0	1	1	1	0	0	0	0
	1	1	1	0	0	0	0	0
	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	1	0	1	1	0	1	0	0
0	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0

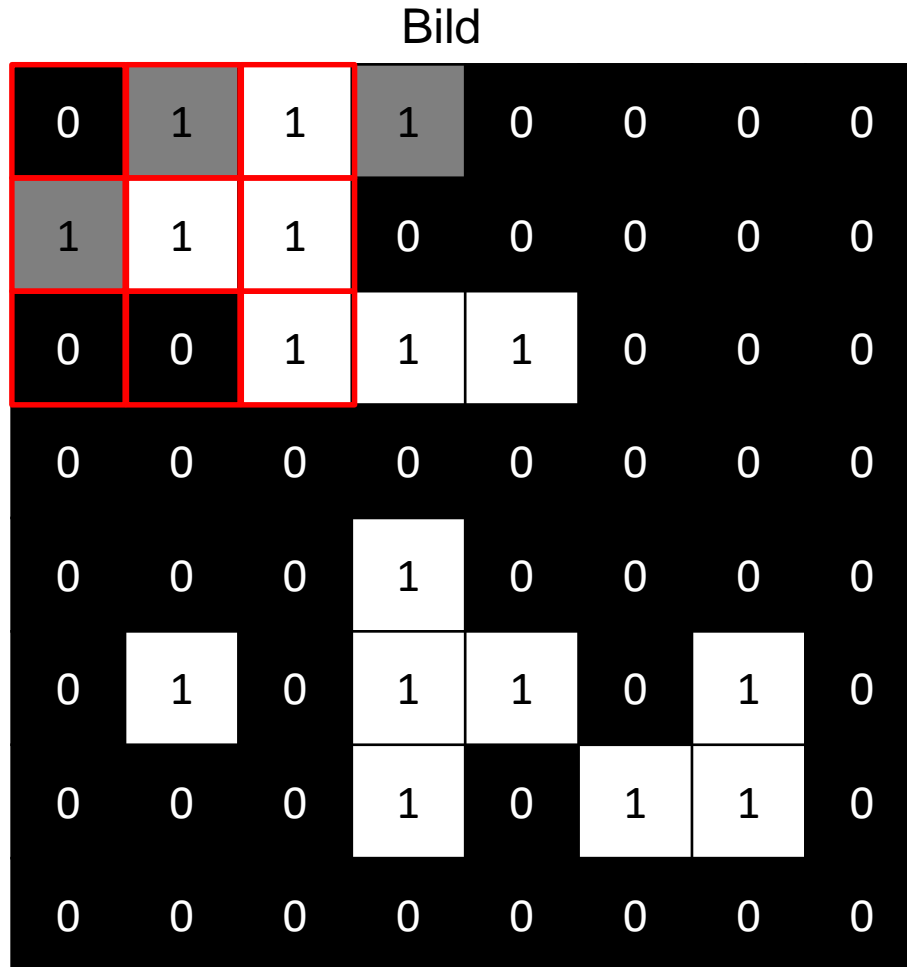
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Falls irgendwo eine 1 im Strukturelement auf eine 1 im Bild trifft, markiere den zentralen Pixel mit 1, sonst mit 0.

Randbedingungen: Alles 0!

Kurze Möglichkeit auf die Formale Definition zurückzukommen

Formale Definition:

Für $G \subset \mathbb{Z}^2$ und $S \subset \mathbb{Z}^2$ ist die Dilatation von G mit S gegeben durch

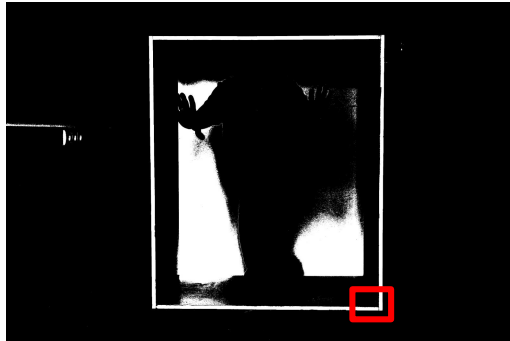
$$G \oplus S := \{x \mid G \cap S_x \neq \emptyset\}$$

wobei

$$S_x := \{x + y \mid y \in S\}$$

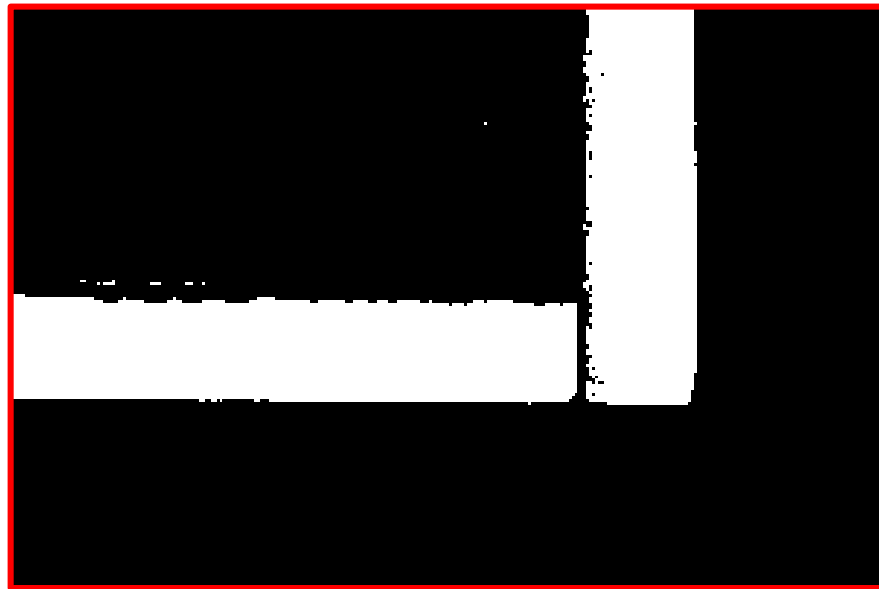
die Verschiebung von S nach x beschreibt.

Wie hilft uns dies weiter?



```
import scipy.ndimage.morphology as morph
a = np.zeros((9,9), dtype=int)
a[2,0:9] = 1;
dilatedZoom = morph.binary_dilation(zoom,a)
```

Unsere problematische Ecke



Nach der Dilatation



Gewisse Ungenauigkeit in dem Bild nach der Dilatation: Die Kanten der Kiste sind entsprechend des Strukturelementes breiter geworden!

Gegenmaßnahme: Das Gegenstück (*duale*) zur Dilatation – die **Erosion**!



Formale Definition:

Für $G \subset \mathbb{Z}^2$ und $S \subset \mathbb{Z}^2$ ist die Erosion von G mit S gegeben durch

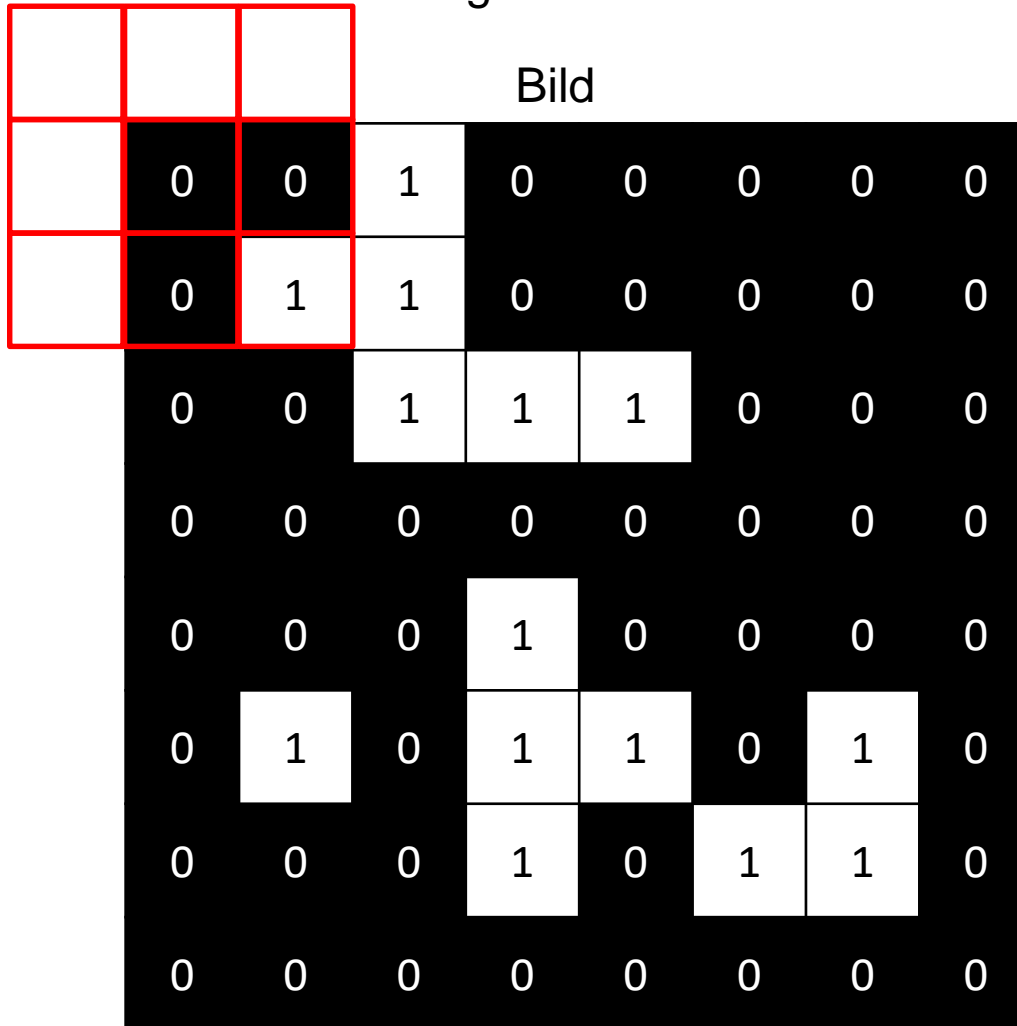
$$G \ominus S := \{x \mid S_x \subset G\}$$

wobei

$$S_x := \{x + y \mid y \in S\}$$

die Verschiebung von S nach x beschreibt.

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	1	0	1	1	0	1	0	0
0	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0

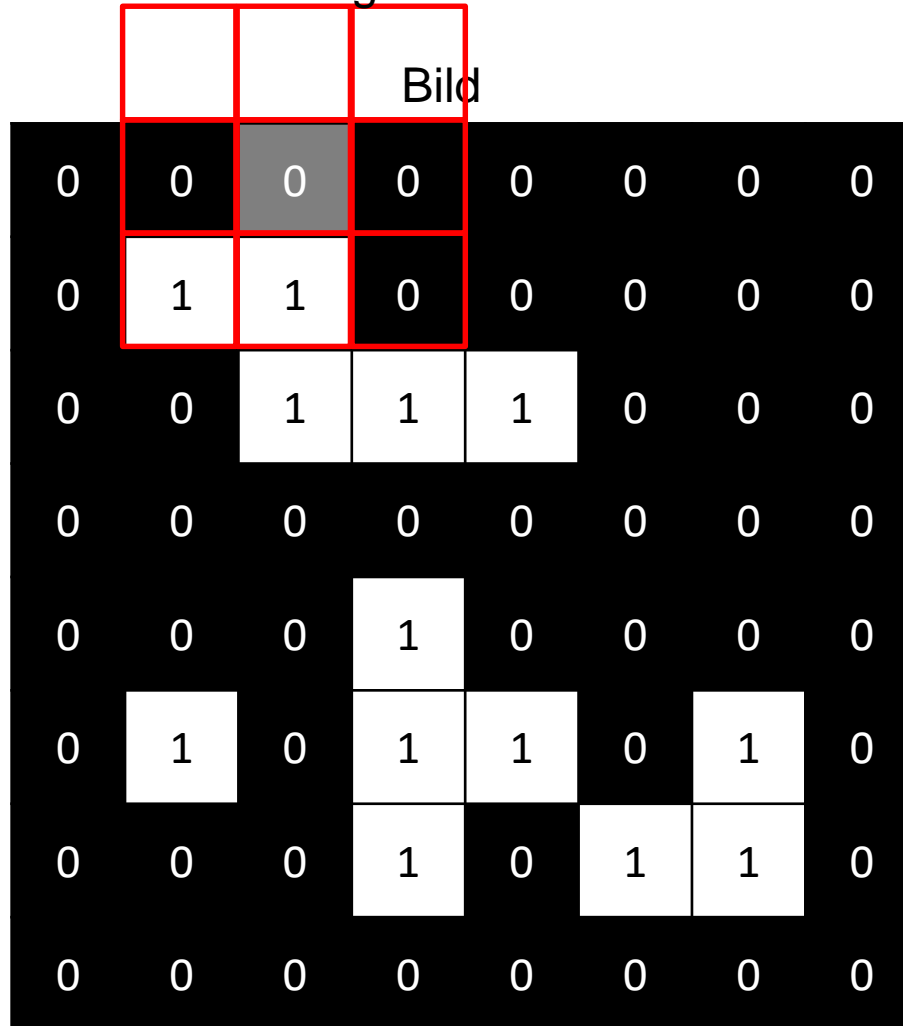
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



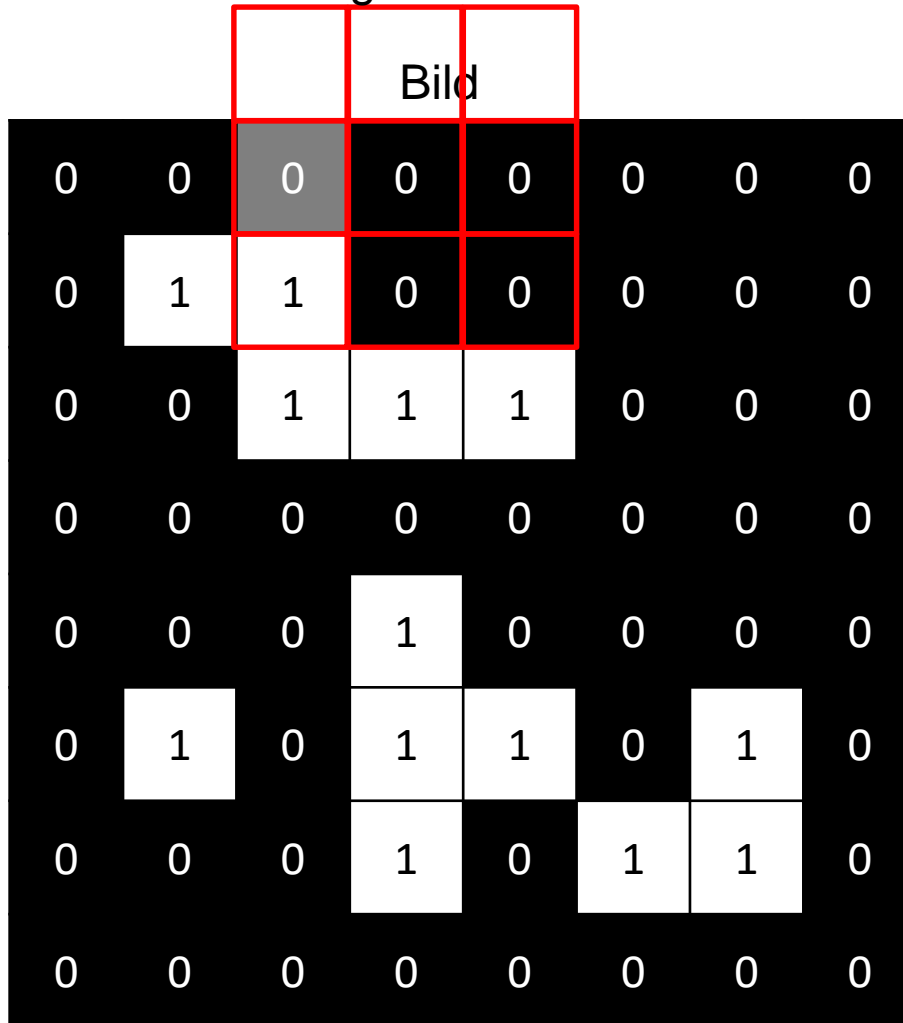
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

	0	0	0	0	0	0	0	0
	0	1	1	0	0	0	0	0
	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	1	0	1	1	0	1	0	0
0	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

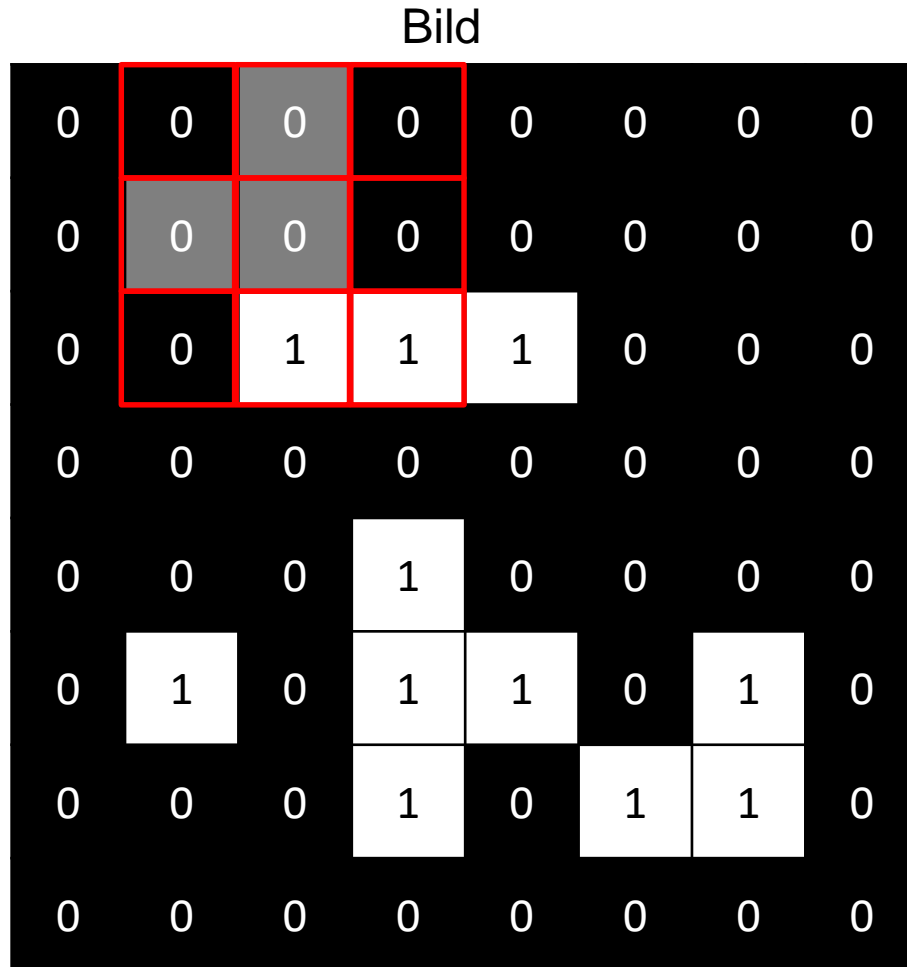
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!



Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0
0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	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	0	0	0	0	0

Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

Wie bei einer Faltung: Schiebe die Maske des Strukturelementes über das Bild!

Bild

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	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	0	0	0	0	0

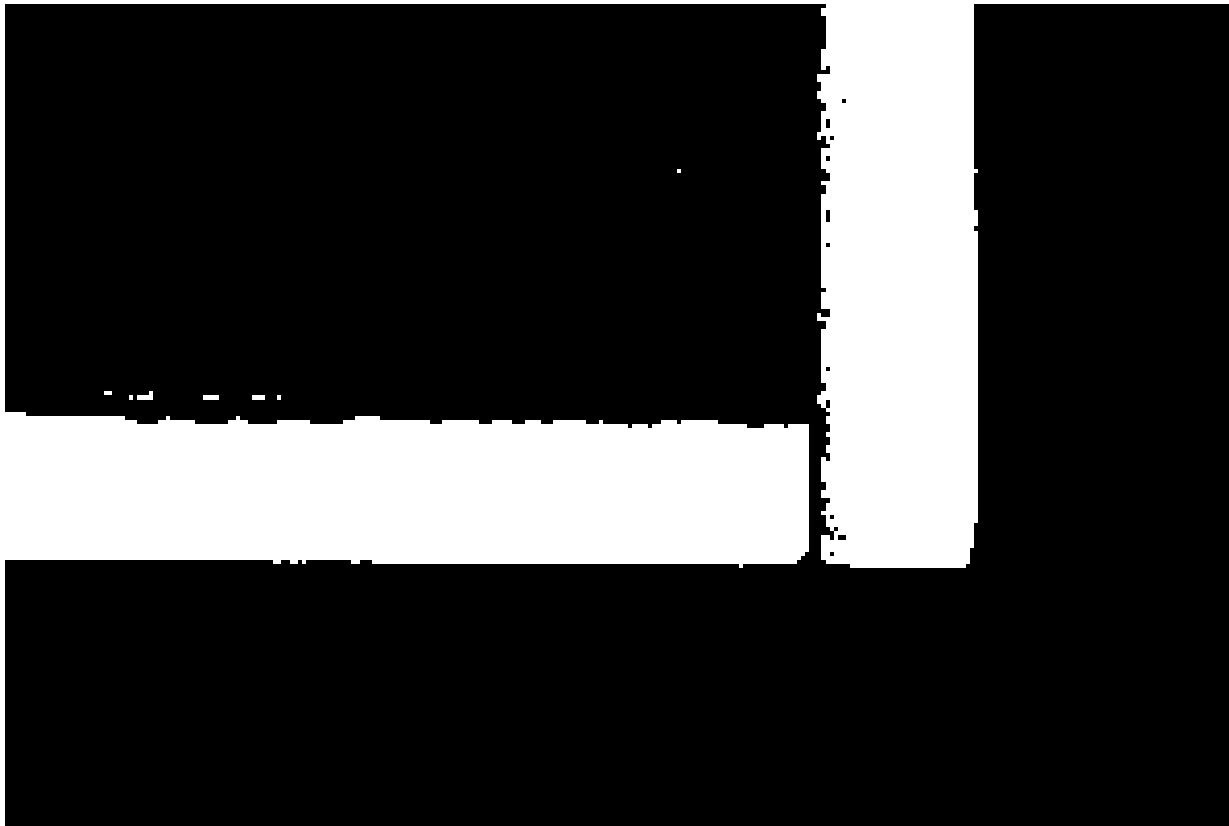
Strukturelement

0	0	0
1	1	1
0	0	0

Regel: Nur wenn alle 1en im Strukturelement auch auf 1en im Bild treffen, ist der zentrale Pixel 1, sonst 0.

Randbedingungen: Alles 0!

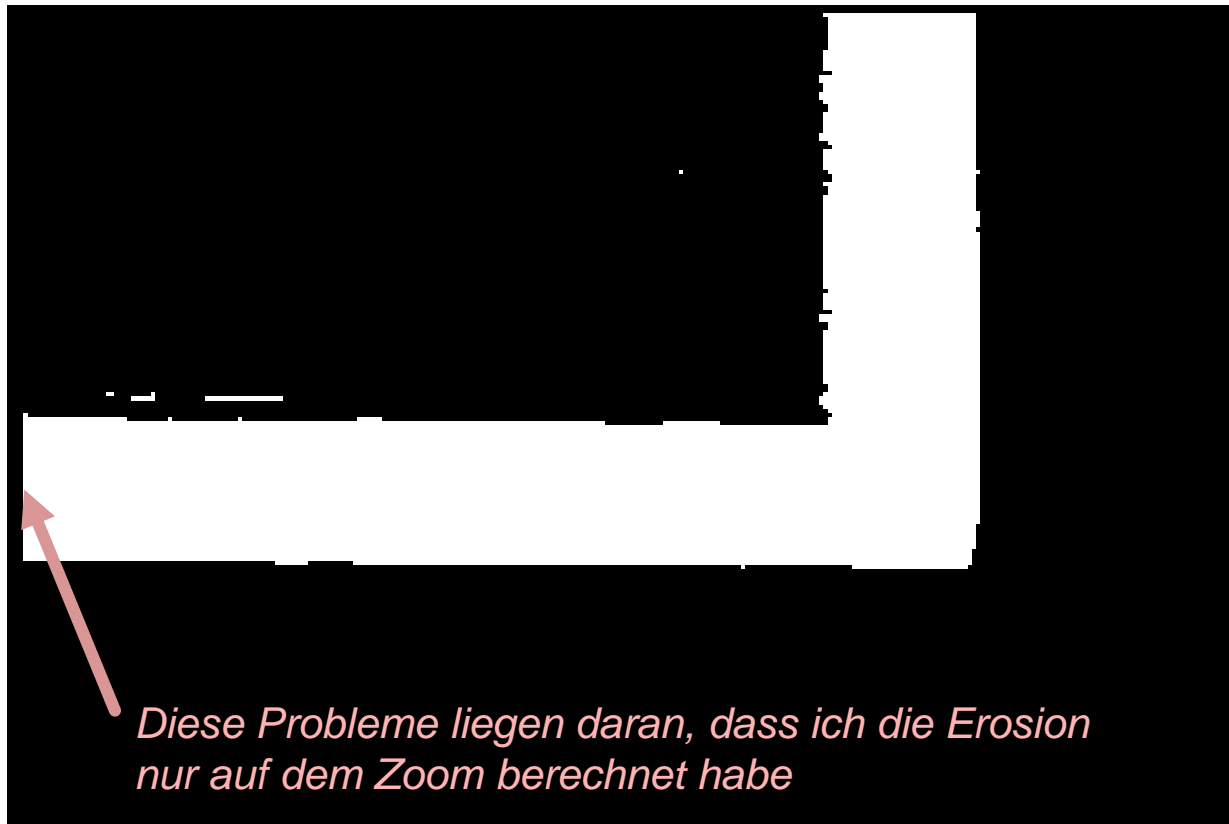
Problemecke



Problemecke nach Dilatation



Problemecke nach Dilatation und anschließender Erosion



Die Erosion der Dilatation eines Bildes mit jeweils dem gleichen Strukturelement nennt man **morphological Closing**

Eigenschaften:

- Bis auf Randeffekte ist die weiße Region des Ergebnisses größer als vor dem Closing
- Bis auf Randeffekte enthält die weiße Region des Ergebnisses die ursprüngliche weiße Region als Teilmenge
- Mehrfaches Anwenden des Closings (mit gleichem Strukturelement) ändert nichts im Vergleich zum einmaligen Anwenden (idempotent)

Genereller Effekt: Es werden Löcher geschlossen in die das Strukturelement nicht hereinpasst ohne das Binärbild ansonsten zu sehr zu verändern!

Das Gegenstück zum Closing ist das **morphological Opening**. Es ist das Anwenden der Dilatation auf das Ergebnis der Erosion eines Bildes.

Eigenschaften:

- Die weiße Region des Ergebnisses kleiner als vor dem Opening
- Die weiße Region des Ergebnisses ist eine Teilmenge der ursprünglichen weißen Region
- Mehrfaches Anwenden des Openings (mit gleichem Strukturelement) ändert nichts im Vergleich zum einmaligen Anwenden (idempotent)

Genereller Effekt: Es werden weiße Bereiche gelöscht in die das Strukturelement nicht hereinpasst ohne das Binärbild ansonsten zu sehr zu verändern!

Hier eine mögliche Segmentierungspipeline



1. Umwandeln in Graustufen



2. Thresholden: Annahme – der Bereich der uns interessiert ist heller als der Rest



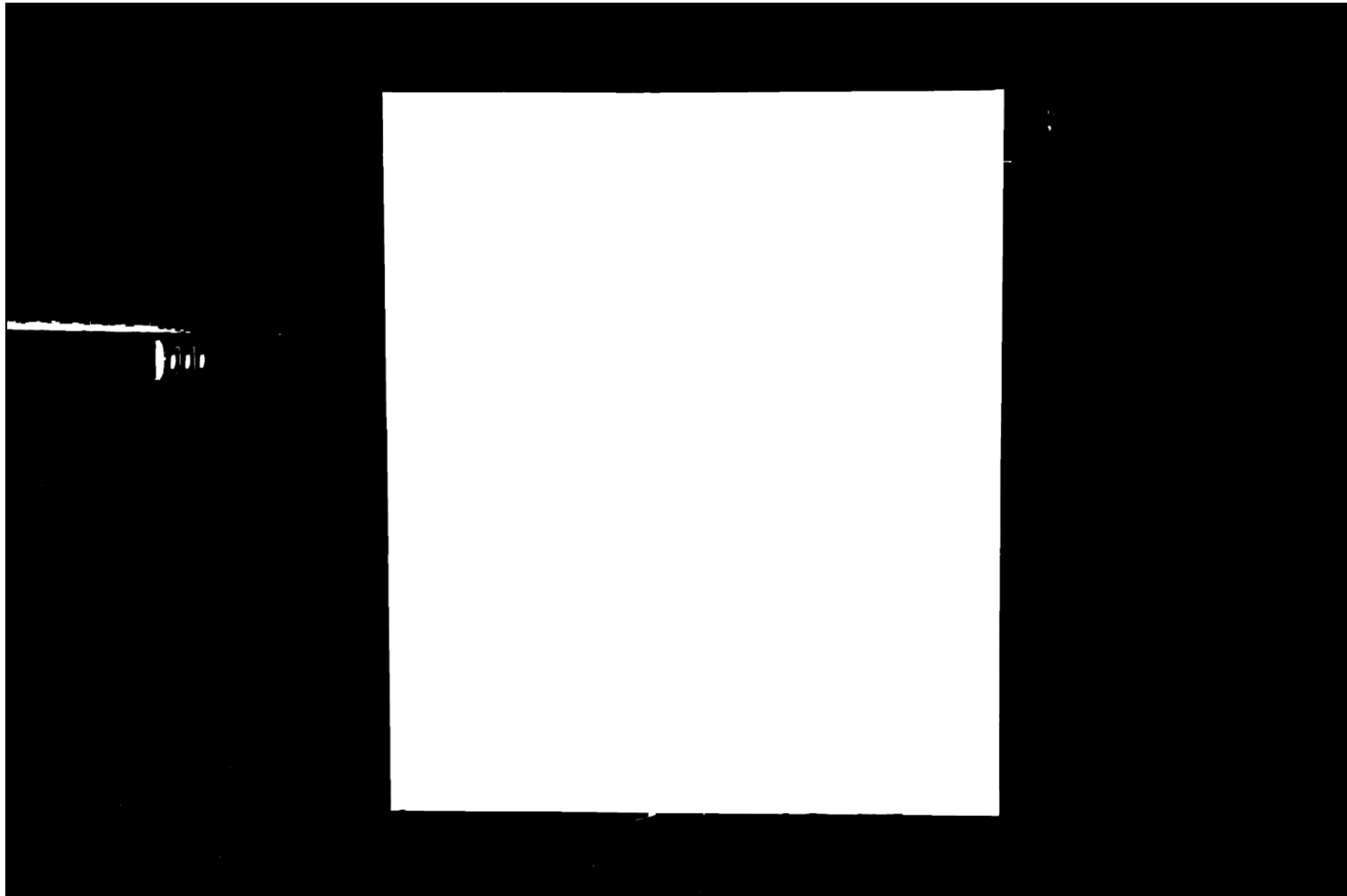
3. Closing mit vertikaler Linie als Strukturelement



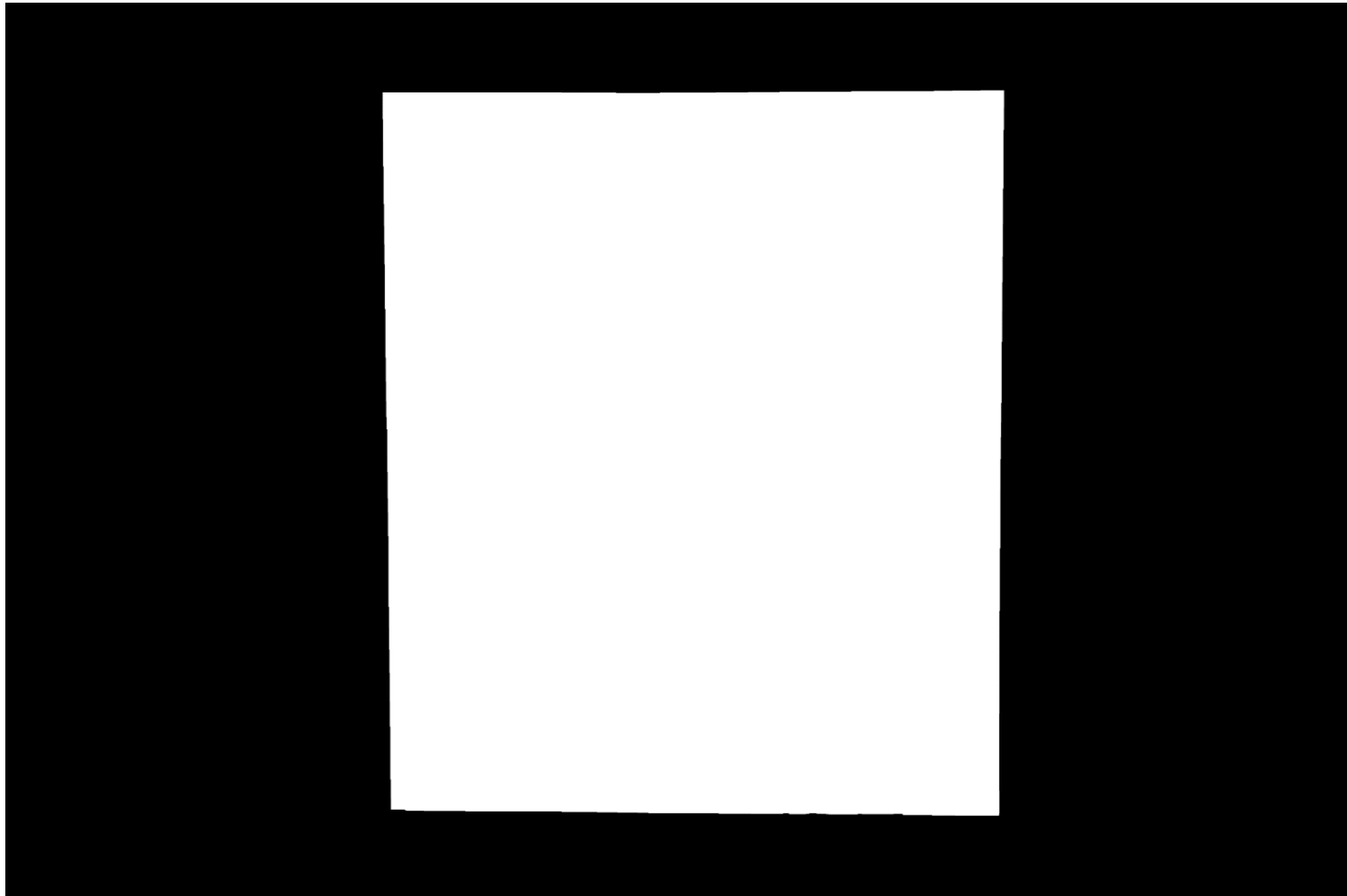
4. Closing mit horizontaler Linie als Strukturelement



5. Füllen geschlossener Bereiche



6. Opening mit quadratischem Strukturelement um unnötige Strukturen zu entfernen

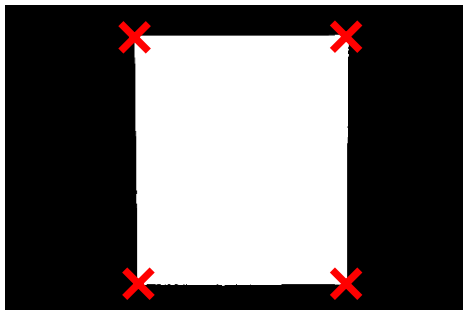


Danach sicherheitshalber noch die größte Region
(Wegzusammenhangskomponente) auswählen

```
from skimage import measure

def getLargestCC(segmentation):
    labels = measure.label(segmentation)
    assert( labels.max() != 0 ) # assume at least 1 CC
    largestCC = labels == np.argmax(np.bincount(labels.flat)[1:])+1
    return largestCC

finalBinary = getLargestCC(binaryImg)
```



Nächster Diskussionspunkt: Wie kommen wir
nun an die Koordinaten der Ecken?