

**Convolution Filter** Ein Convolution Filter wird für die (künstliche bzw. nachträgliche) Reduktion oder Erhöhung der Schärfe von Bildern verwendet. Das Eingabebild wird als Matrix von Farb- und Helligkeitswerten interpretiert. Ein sog. Kernel, eine kleine benutzerdefinierte Matrix, wird schrittweise „über“ das Bild geschoben. Jeder Pixel nimmt einen Wert in Abhängigkeit von benachbarten Pixeln, skaliert durch die Werte des Kernels, an.

Bei einem Kernel von

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

...und einem Bild das in Matrixform als

$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

dargestellt wird,

würde der Kernel nun über die Bildmatrix geschoben, wobei jeweils übereinanderliegende Elemente miteinander multipliziert werden. Der Bildpixel in der Mitte des Kernels nimmt nun den Durchschnitt dieser Produkte an.

$$\begin{array}{cccccc}
\theta & \theta & \theta & 0 & 0 & 0 \\
\theta & \boxed{1} & 1 & 1 & 1 & 0 \\
\theta & 1 & \theta & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\quad
\begin{array}{cccccc}
\theta & \theta & \theta & \theta & 0 & 0 \\
0 & 1 & \boxed{1} & 1 & 1 & 0 \\
0 & 1 & \theta & \theta & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\quad
\begin{array}{cccccc}
0 & 0 & \theta & \theta & \theta & 0 \\
0 & 1 & 1 & \boxed{1} & 1 & 0 \\
0 & 1 & \theta & \theta & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\quad \dots$$

Die jeweiligen Ergebniswerte im Beispiel sind  $\frac{1}{3}$ ,  $\frac{4}{9}$  und  $\frac{4}{9}$ . Dieser Prozess wird für alle Elemente der Bildmatrix wiederholt.

Beispiel angelehnt an <https://iq.opengenus.org/convolution-filters/>.

**High- und Low-Resolution** Die Auflösung eines Bilds im Rasterformat ist definiert durch die Anzahl der Pixel die das Bild darstellen. Ein Bild mit einer Auflösung von 5000x5000 ist hochauflösender als ein Bild mit einer Auflösung von 1000x1000.

Bei der Einbindung von Bildern in Webseiten ist es oft sinnvoll die Auflösung so niedrig wie möglich zu wählen. So sind niedrigauflösende Bilder gut geeignet als Thumbnails. Hier fällt die niedrige Auflösung bei den üblicherweise kleinen Darstellungsgrößen nicht sonderlich auf, aber die Dateigröße kann drastisch kleiner sein. Bei Bildern die in voller Bildschirmgröße angezeigt werden, kann es sinnvoll sein das Bild in der Bildschirmauflösung (also üblicherweise recht hochauflösend) anzuzeigen um die bestmögliche Qualität zu erreichen.

Das wird auch in Zukunft sinnvoll bleiben, keine unnötig grossen Daten zu laden ist immer gut.

**Aliasing-Effekte** Aliasing-Effekte treten in Bildern auf wenn die Sampling-Rate zu hoch ist, wenn also die Auflösung unter einem gewissen kritischen Punkt ist. Dann erscheinen etwa eigentlich unterschiedlich aussehende Elemente als gleich, wie im folgenden Beispiel erkenntlich ist.



Beispielbilder entlehnt aus <https://en.wikipedia.org/wiki/Aliasing>.

## JPEG

- a) Es folgt eine grobe Erläuterung der Kompressionsschritte des JPEG-Formats.
  - 1) Farbtransformation in YCbCr (Y = Luminanz, CbCr = Chrominanz — blau und rot) Farbraum
  - 2) Downsampling der Chrominanzinformationen.
  - 3) Diskrete Kosinustransformation in 8x8 Blöcke.
  - 4) Normalisierung der durch DCT erhaltenen Blöcke anhand einer Normalisierungstabelle die bei Kompression anzugeben ist und im komprimierten Bild enkodiert ist. (Elementweise Division.)
  - 5) Kodierung der nun erhaltenen Daten, etwa durch Huffman-Kodierung.
- b) Nach der diskreten Kosinustransformation und Normalisierung haben Zellen mit niedrigem Spalten- bzw. Zeilenindex üblicherweise höhere Werte als solche mit hohen Indices. Bei hohen Spalten- und Zeilenindices sammeln sich Nullen. Durch das „zigzag“ Auslesen sammeln sich die Nullen am Ende, damit verbessert sich die Komprimierbarkeit der Daten im Vergleich zum vertikalen oder horizontalen Auslesen.
- c) Verlustbehaftete Kodierung wegen der Normalisierung.