Lesson problems: Convolution

1 1D discrete convolution

a) Calculate, step by step by hand, the convolution

$$g(x) = (h * f)(x) = \sum_{\lambda=-\infty}^{\infty} h(x - \lambda) f(\lambda),$$

where $f(x) = (..., 0, 0, 1, -1, -2, 0, -1, 1, 2, -1, 0, 0, ...)$ is the 1D signal and $h(x) = (..., 0, 0, 1, 2, -2, 0, 0, ...)$ is the impulse response, i.e. the 1D convolution kernel. These functions can be represented by the following vectors:

$f(x) =$

1	-1	-2	**0**	-1	1	2	-1

and

$h(x) =$

1	**2**	-2

,

where the respective numbers written in bold face are at position $x = 0$. All positions in f(x) and h(x) outside the vectors are equal to zero.

b) Show how the convolution also can be obtained through a matrix multiplication.

2 2D discrete convolution

Calculate, step by step by hand, the convolution

$$g(x, y) = (h * f)(x, y) = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} h(x - \alpha, y - \beta) f(\alpha, \beta),$$

where

$f(x, y) =$

1	**2**	3
1	1	1

is the 2D signal and $h(x, y) =$

1	**3**
2	0

is the impulse response, i.e. the 2D convolution kernel. In the matrices, the respective numbers written in bold face are at position $(x, y) = (0, 0)$. All positions in $f(x, y)$ and $h(x, y)$ outside the matrices are equal to zero.

3 2D convolution and correlation applied on images

The image $f(x, y)$ on the next page consists of a small 4x4-square of eights "8". Each empty position has value 0. Positions outside the image are omitted, i.e. when calculating the convolution, those positions can be considered to have value 0. (Here, it is not necessary to define the position $(x, y) = (0, 0)$.)

		8	8	8	8		
		8	8	8	8		
		8	8	8	8		
		8	8	8	8		

f(x,y)

gx(x,y)

gy(x,y)

a) In the convolution kernels dx and dy below, the respective bold face center digit "0" is at position $(x, y) = (0, 0)$.

$$dx = \begin{array}{|c|c|c|}\hline 0.5 & \mathbf{0} & -0.5 \\ \hline\end{array} = \begin{array}{|c|c|c|}\hline 1 & \mathbf{0} & -1 \\ \hline\end{array} \;/\; 2, \qquad dy = \begin{array}{|c|}\hline -0.5 \\ \hline \mathbf{0} \\ \hline 0.5 \\ \hline\end{array} = \begin{array}{|c|}\hline -1 \\ \hline \mathbf{0} \\ \hline 1 \\ \hline\end{array} \;/\; 2$$

Calculate, step by step by hand, the two convolutions

$$gx(x, y) = (dx * f)(x, y) = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} dx(x - \alpha, y - \beta) f(\alpha, \beta),$$

$$gy(x, y) = (dy * f)(x, y) = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} dy(x - \alpha, y - \beta) f(\alpha, \beta).$$

Write the results in the two respective matrices $gx(x, y)$ and $gy(x, y)$ given in the figure above.

b) Explain from your results how the convolutions with the kernels dx and dy are equivalent to derivation of the image $f(x, y)$ in the x- and y-direction, respectively.

c) The kernels $vert$ and $horis$ below are equivalent to a 180 degree rotation of the kernels dx and dy, respectively.

$$vert = \begin{array}{|c|c|c|}\hline -1 & \mathbf{0} & 1 \\ \hline\end{array} \;/\; 2, \qquad horis = \begin{array}{|c|}\hline 1 \\ \hline \mathbf{0} \\ \hline -1 \\ \hline\end{array} \;/\; 2$$

Calculate, step by step by hand, the two **correlations**

$$g_{vert}(x,y) = (vert\Box f)(x,y) = \sum_{\alpha=-\infty}^{\infty}\sum_{\beta=-\infty}^{\infty} vert(\alpha-x,\beta-y)f(\alpha,\beta),$$

$$g_{horis}(x,y) = (horis\Box f)(x,y) = \sum_{\alpha=-\infty}^{\infty}\sum_{\beta=-\infty}^{\infty} horis(\alpha-x,\beta-y)f(\alpha,\beta).$$

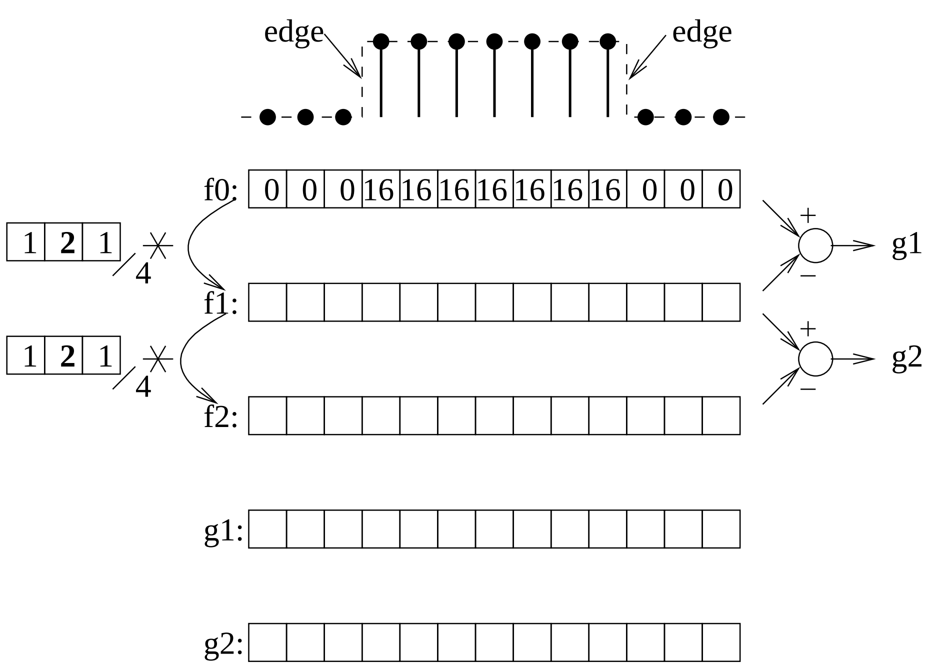
The symbol $\Box$ denotes correlation.

d) Explain how can you see in your results that the correlations with $vert$ and $horis$ provide matches for vertical and horisontal edges, respectively, in the image.

4 1D lowpass and bandpass filtering

The pixel values along a horisontal row in an image can be considered as a one dimensional discrete function. Below is one such horisontal row, with pixel values $(..., 0, 0, 0, 16, 16, 16, 16, 16, 16, 16, 0, 0, 0, ...)$, which can be interpreted as a sampled discrete rectangle function, see the top figure below. We assume that all pixel values to the left and to the right of the sequence are zero.

A convolution of this discrete function and the kernel $(1/4, 2/4, 1/4) = (1, 2, 1)/4$ is equal to a low pass filtering of the function, which in turn is equal to blurring along the image row. The bold face "2" is at position 0, as above.



a) Perform the two-step lowpass filtering (blurring) with the convolution kernel, according to the figure above. Describe in words how the "edges" of the rectangular function are affected by the lowpass filter efter each convolution.

b) The output $g1$ of the top subtraction in the figure is equivalent to a highpass filtering of the function $f0$ (= the image row) and the output $g2$ of the bottom subtraction in the figure is equivalent to a bandpass filtering of the function $f0$.

   - Describe in words how smooth surfaces along an image row is affected by a highpass filter and by a bandpass filter.
   - Describe in words how edges along an image row is affected by a highpass filter and by a bandpass filter.

c) Small box filter kernels $(1, 1)/2$ can be used to construct larger filter kernels. Check that

$$\begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 = \begin{array}{|c|c|c|}\hline 1 & \mathbf{2} & 1\\\hline\end{array}/4.$$

The center of the box filter is beteen the two pixels, here denoted with a double line. Similarly as before, the pixel with the bold face "2" is at position 0. Then compute

$$\begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2$$

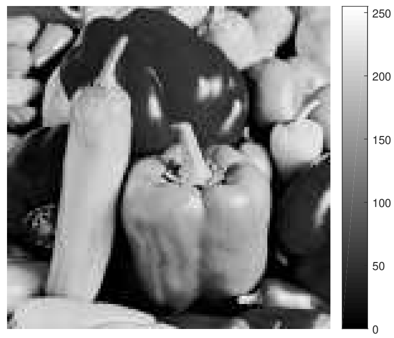
and

$$\begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2\,.$$

d) The small box filter kernel $(1, 1)/2$ and its transpose $(1, 1)'/2$ can be used to construct larger 2D filter kernels. Compute

$$\begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c||c|}\hline 1 & 1\\\hline\end{array}/2 * \begin{array}{|c|}\hline 1\\\hline\hline 1\\\hline\end{array}/2 * \begin{array}{|c|}\hline 1\\\hline\hline 1\\\hline\end{array}/2\,.$$

5 The effect of convolution kernels on 2D images

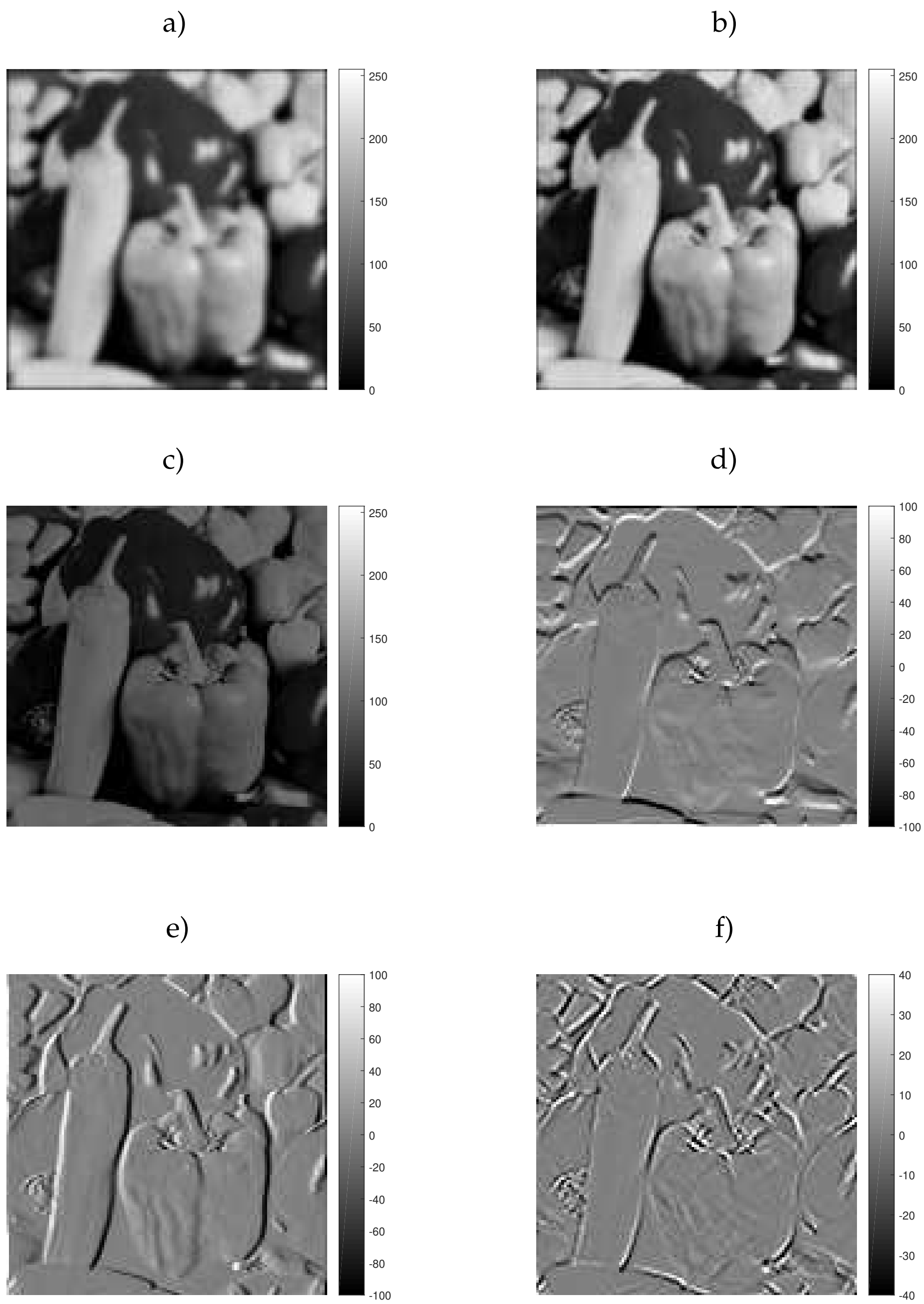


On the next page are the results after the convolutions of the original image above with 6 different convolution kernels A, B, C, D, E=A*A*A, and F (an unknown kernel), where

$$A=\begin{array}{|c|}\hline 1\\\hline \mathbf{2}\\\hline 1\\\hline\end{array}/4 * \begin{array}{|c|c|c|}\hline 1 & \mathbf{2} & 1\\\hline\end{array}/4 = \begin{array}{|c|c|c|}\hline 1 & 2 & 1\\\hline 2 & \mathbf{4} & 2\\\hline 1 & 2 & 1\\\hline\end{array}/16, \qquad B=\begin{array}{|c|c|c|}\hline 1 & \mathbf{0} & -1\\\hline\end{array}/2,$$

$$C=\begin{array}{|c|}\hline -1\\\hline \mathbf{0}\\\hline 1\\\hline\end{array}/2, \qquad D=\begin{array}{|c|c|c|}\hline 0 & 0 & 0\\\hline 0 & \mathbf{1} & 0\\\hline 0 & 0 & 0\\\hline\end{array}/2.$$

Match each resulting image with its corresponding convolution kernel and give a clear motivation for each match.


a)
b)
c)
d)
e)
f)
250
200
150
100
50
0
100
80
60
40
20
0
-20
-40
-60
-80
-100
40
30
20
10
0
-10
-20
-30
-40

6 Downsampling

Downsampling by a factor 2 is traditionally carried out by convolving the image with a low pass filtering (blurring) kernel, like for example

$$f = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & \mathbf{4} & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} /16,$$

followed by a step where every second pixel along both each row and each column are discarded. Finally the image is reshaped to the resulting lower resolution image. (The low pass filtering is needed to avoid so called aliasing.)

a) Perform by hand a downsampling by a factor 2 of the image in the figure below. Empty pixels and "pixel positions" outside the image have value 0.

b) The stride downsampling method is popular in the field of deep learning. Describe in words how, using this method, you can go directly from the original image to the resulting downsampled image, without performing the intermediate steps.