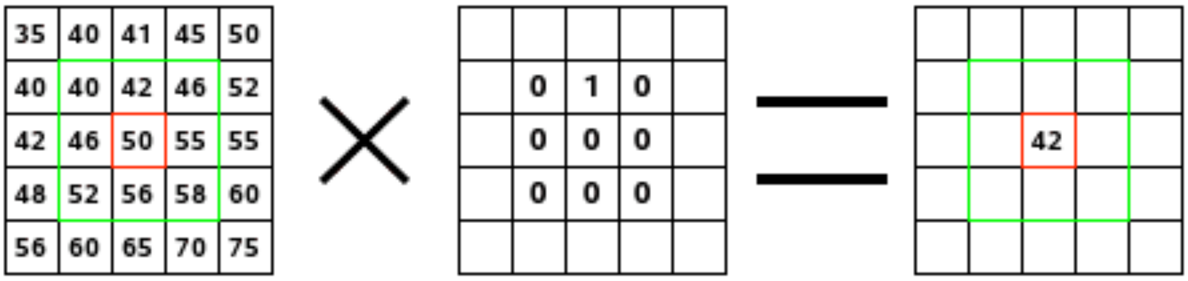
8.2. Convolution Matrix

8.2.1. Overview

Here is a mathematician's domain. Most of filters are using convolution matrix. With the Convolution Matrix filter, if the fancy takes you, you can build a custom filter.
What is a convolution matrix? It's possible to get a rough idea of it without using mathematical tools that only a few ones know. Convolution is the treatment of a matrix by another one which is called “kernel”.
The Convolution Matrix filter uses a first matrix which is the Image to be treated. The image is a bi-dimensional collection of pixels in rectangular coordinates. The used kernel depends on the effect you want.
GIMP uses 5x5 or 3x3 matrices. We will consider only 3x3 matrices, they are the most used and they are enough for all effects you want. If all border values of a kernel are set to zero, then system will consider it as a 3x3 matrix.
The filter studies successively every pixel of the image. For each of them, which we will call the “initial pixel”, it multiplies the value of this pixel and values of the 8 surrounding pixels by the kernel corresponding value. Then it adds the results, and the initial pixel is set to this final result value.
A simple example:



35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

×

	0	1	0	
	0	0	0	
	0	0	0	

=

		42		

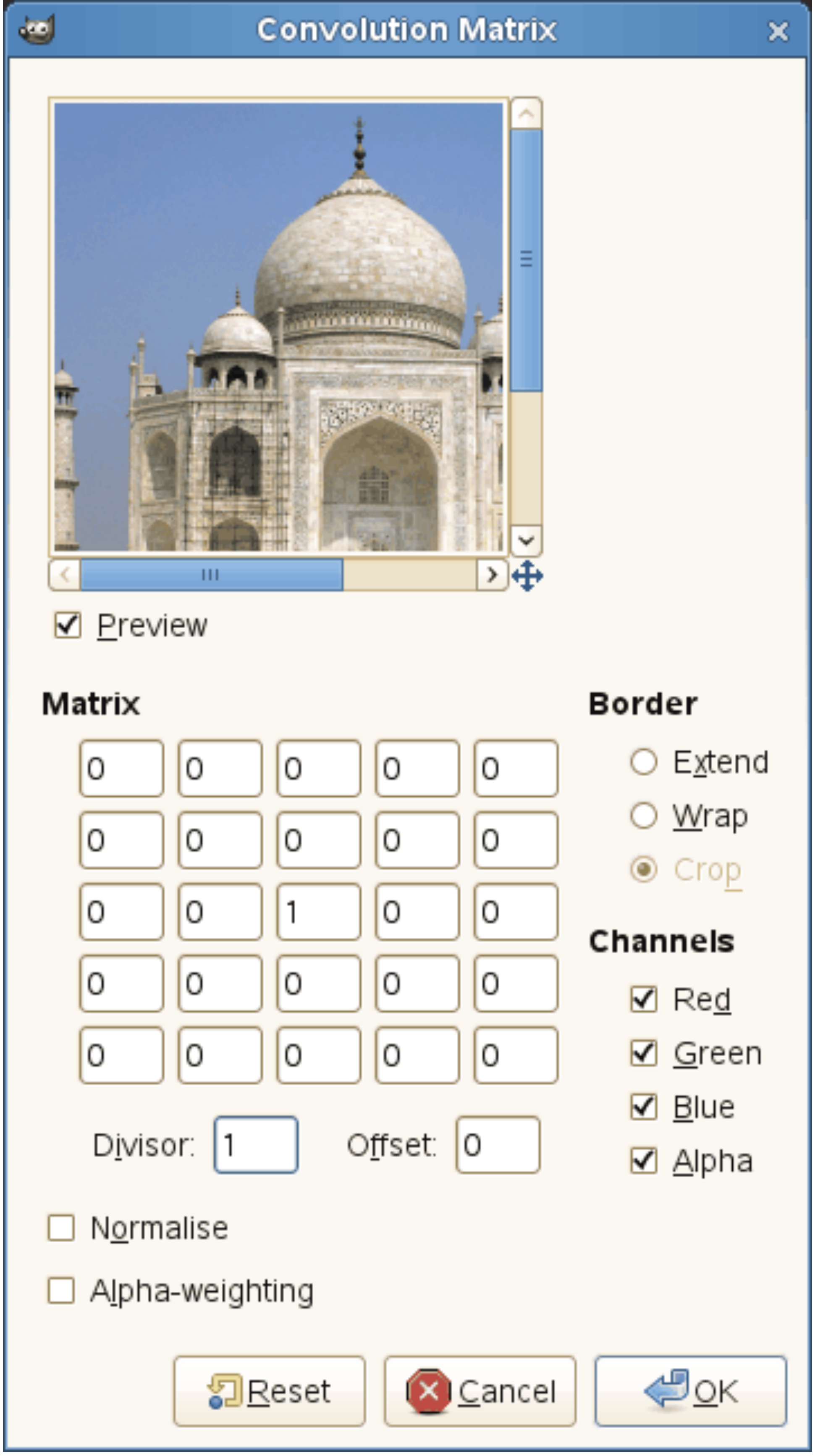
On the left is the image matrix: each pixel is marked with its value. The initial pixel has a red border. The kernel action area has a green border. In the middle is the kernel and, on the right is the convolution result.
Here is what happened: the filter read successively, from left to right and from top to bottom, all the pixels of the kernel action area. It multiplied the value of each of them by the kernel corresponding value and added results. The initial pixel has become 42: (40*0)+(42*1)+(46*0) + (46*0)+(50*0)+(55*0) + (52*0)+(56*0)+(58*0) = 42. (the filter doesn't work on the image but on a copy). As a graphical result, the initial pixel moved a pixel downwards.

8.2.2. Activating the filter

This filter is found in the image window menu under Filters → Generic → Convolution Matrix….

8.2.3. Options

Figure 17.150. “Convolution matrix” options



☑ Preview

Matrix

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

Divisor: 1 Offset: 0

□ Normalise

□ Alpha-weighting

Border

○ Extend

○ Wrap

● Crop

Channels

☑ Red

☑ Green

☑ Blue

☑ Alpha

Matrix

This is the 5x5 kernel matrix: you enter wanted values directly into boxes.

Divisor

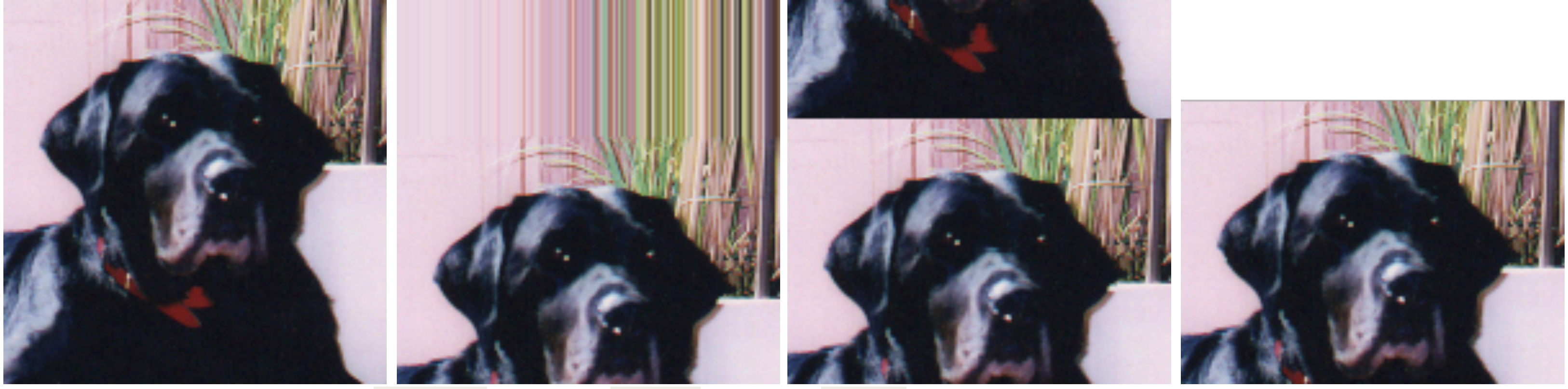
The result of previous calculation will be divided by this divisor. You will hardly use 1, which lets result unchanged, and 9 or 25 according to matrix size, which gives the average of pixel values.

Offset

This value is added to the division result. This is useful if result may be negative. This offset may be negative.

Border

When the initial pixel is on a border, a part of kernel is out of image. You have to decide what filter must do:



From left: source image, Extend border, Wrap border, Crop border

Extend

This part of kernel is not taken into account.

Wrap

This part of kernel will study pixels of the opposite border, so pixels disappearing from one side reappear on the other side.

Crop

Pixels on borders are not modified, but they are cropped.

Channels

You can select there one or several channels the filter will work with.

Normalise

If this option is checked, The Divisor takes the result value of convolution. If this result is equal to zero (it's not possible to divide by zero), then a 128 offset is applied. If it is negative (a negative color is not possible), a 255 offset is applied (inverts result).

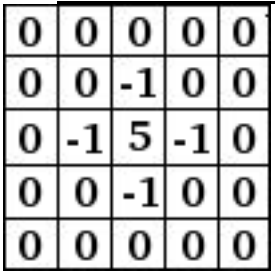
Alpha-weighting

If this option is not checked, the filter doesn't take in account transparency and this may be cause of some artefacts when blurring.

8.2.4. Examples

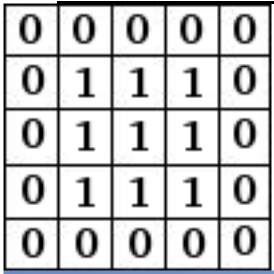
Design of kernels is based on high levels mathematics. You can find ready-made kernels on the Web. Here are a few examples:

Figure 17.151. Sharpen



0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

Figure 17.152. Blur



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

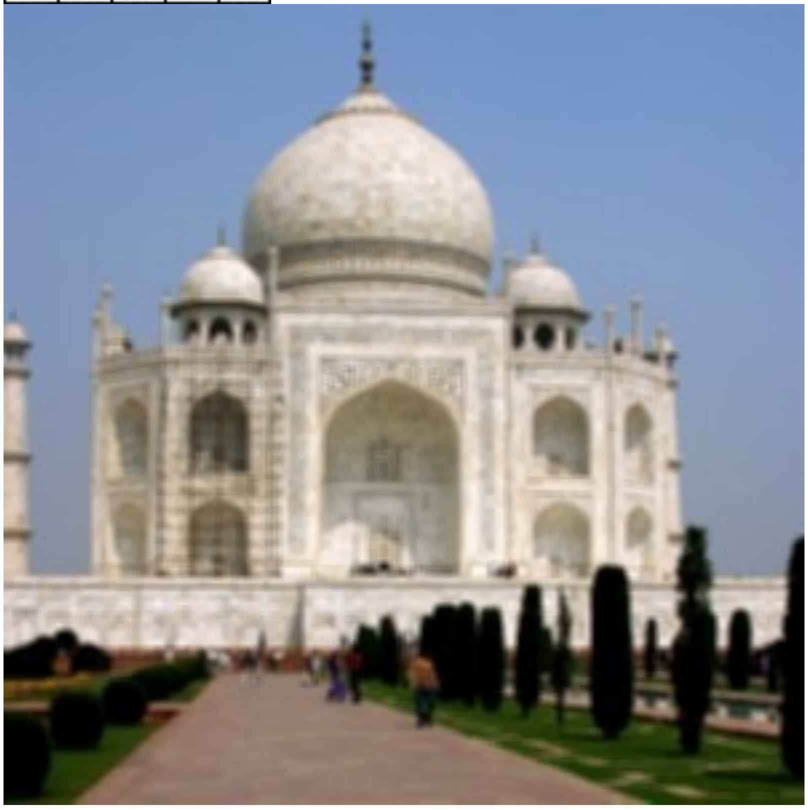
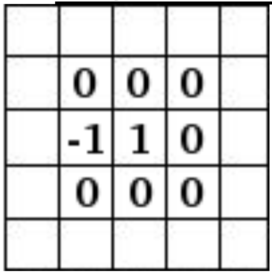


Figure 17.153. Edge enhance



	0	0	0	
	-1	1	0	
	0	0	0	

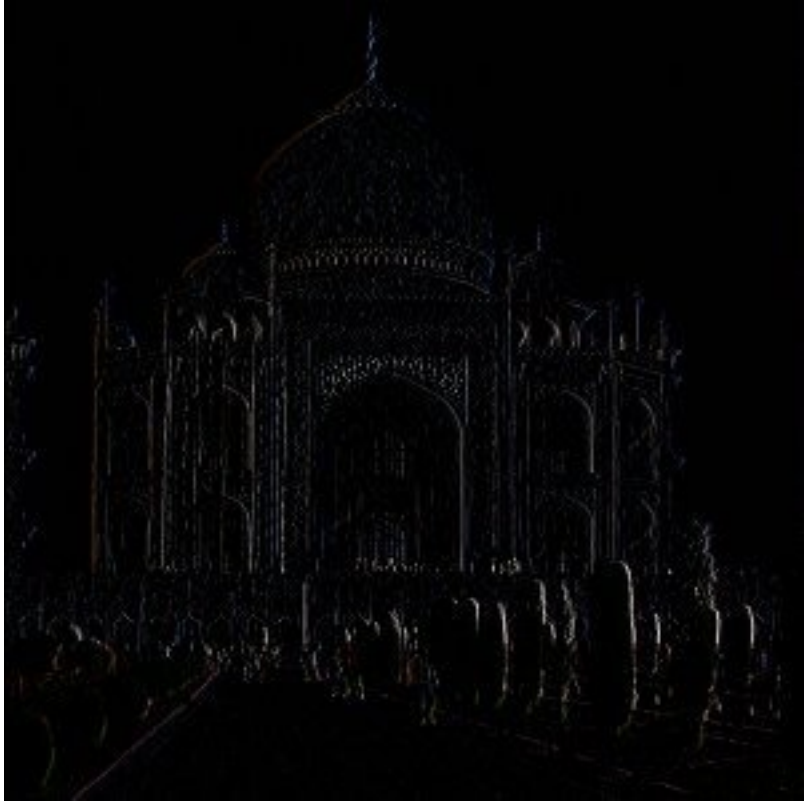


Figure 17.154. Edge detect

	0	1	0	
	1	-4	1	
	0	1	0	

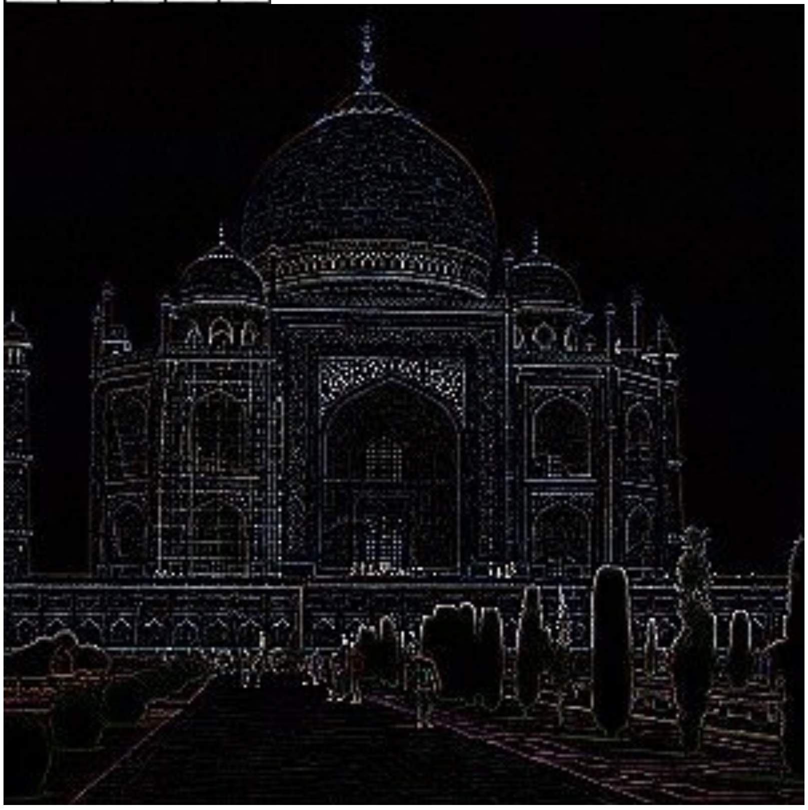


Figure 17.155. Emboss

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