volution

The effect of this operation is that a template, $t(x,y)$, is placed **over all possible image locations**. At each location, the product of an image value and the overlapping template value is computed. The products are summed to give the output value at that location. It requires n^2 multiplications and additions per pixel, where n is the size of the template. We need to **normalise** the output. (divide the sum by n or convolve with $\frac{1}{n}$, $n \times n$ matrix)

Image Pixel

i_a	i_b	i_c
i_d	i_e	i_f
i_g	i_h	i_i

Convolution Matrix

c_a	c_b	c_c
c_d	c_e	c_f
c_g	c_h	c_i

$$i_e = \sum_{k=a}^{i} i_k \cdot c_k$$. After this we need to normalise it

$$\Downarrow \quad \frac{i_e}{i}$$

Separable templates to convolve with:

1) convolve with $n \times n$ template
   - n^2 multiplications and additions per output pixel

2) convolve with two $n \times 1$ templates
   - $2n$ multiplications and additions per output pixel => faster processing

Where is convolution used?

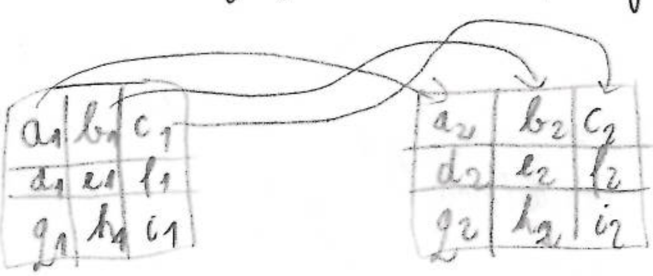
→ **Smoothing** (noise reduction)

→ **Sharpening** (edge enhancement)

→ Template **matching** (finding objects)

Smoothing (noise reduction)

- the **effect** of smoothing on an image is to remove sharp, sudden changes in the brightness function. These might be caused by noise in the image capture device or small objects in the scene that might obscure the larger objects of the scene
- the **result** of smoothing an image is another image with an improved signal to noise ratio, or an image in which the effect of distracting artefacts has been reduced
- **simplest** method is averaging the values after you multiplied them



a_1	b_1	c_1
d_1	e_1	f_1
g_1	h_1	i_1

a_2	b_2	c_2
d_2	e_2	f_2
g_2	h_2	i_2

$$a_1 \cdot a_2 + b_1 \cdot b_2 + c_1 \cdot c_2 + d_1 d_2 + e_1 e_2 + f_1 f_2 + g_1 g_2 + h_1 h_2 + i_1 i_2 = X$$

$\frac{X}{9} \Rightarrow x'$ goes here